

ChatGPT-Resistant Screening Instrument for Identifying Non-Programmers Supplementary Material

1 HOW TO MISLEAD CHATGPT

1. Function Code Generation:

First, it is necessary to define a programming function, denoted by f , that accepts parameter(s) and produces output(s). To simplify it, we almost always use a single input x and a single return value $y = f(x)$. The function f can be created manually with complete control or automatically with ChatGPT by providing desired properties as requirements in a prompt. For example, one can specify the function's desired purpose, its length, or what elements it should include or not include. One can also have different functions f_1, \dots, f_m automatically generated simultaneously, thus discovering new designs and saving time.

2. Input/Output Generation:

Next, we can instruct ChatGPT to generate a set of input-output pairs $(x_1, \hat{y}_1), \dots, (x_n, \hat{y}_n)$ for the desired f . ChatGPT will typically begin with valid examples of non-trivial functions, possibly known from its training data. However, requesting a more significant number of pairs may result in some invalid examples being generated. Of course, we can always automatically increase the number of pairs n by asking for new pairs. In this step, we can also specify desired properties about the form and content of input and output to ChatGPT. For instance, it may be beneficial to restrict the input string's length, ensuring that they are either randomly composed or sourced from an existing dictionary.

3. Invalid Pair Identification:

Now we can run the function for each input x_i proposed by ChatGPT on a dedicated computer and receive the actual output $y_i: y_i = f(x_i) \forall i \in 1, \dots, n$. Then we compare each actual output y_i with the output \hat{y}_i proposed by Chat-GPT. If $y_i \neq \hat{y}_i$, then (x_i, \hat{y}_i) is an invalid input-output pair and thus a possible candidate for a code comprehension question. It may be that such a pair was created only by slight chance, and in most other cases, the correct output is given by ChatGPT. Then the pair would not be suitable after all.

4. Candidate Verification:

Last, we repetitively ask ChaGPT to generate L outputs $\hat{y}_i^{(1)}, \dots, \hat{y}_i^{(L)}$ for a candidate input x_i and the corresponding function f . The number of false outputs from ChatGPT W can be determined by counting the number of times these outputs do not match the correct output ($W = |\{\hat{y}_i^j \mid y_i \neq \hat{y}_i^j \text{ for } j \in \{1, \dots, L\}\}|$). Then we can calculate the $ErrorScore = W/L$ ($ErrorScore \in [0, 1]$). In the best case, the ChatGPT output never matches the correct solution ($ErrorScore = 1$). Otherwise, the relative proportion of errors should be as high as possible. If an attacker manages to transfer a function, including input, into ChatGPT and "execute" it, the probability of success of ChatGPT should be lower than that of guessing.

Notably, certain functions of a relatively simplistic nature may prevent the identification of any invalid input-output pairs in step 2 and must be discarded. In step 3, of course, there may be no invalid pairs. However, by repeating step 2, it may be possible to find some. Only step 4 allows a more precise estimation of whether an x_i together with f consistently yields wrong results and is, therefore, enormously important. If a function of ChatGPT is evaluated repeatedly in the same session, the result will not vary. Therefore it is necessary to start a new session after each "execution" to determine the output distribution and ErrorScore.

Since specifying short return values in the answer options does not involve much cognitive load and time consumption, we increase the number of answer options to eight for these tasks. While we do not burden programmers significantly more, we thus reduce the chance of attackers guessing correctly. We adopt this approach for all code comprehension questions in general since they are also characterized by short answer options.

2 FISHER'S EXACT TEST RESULTS

Table 1. Results of Fisher's exact test for screening questions that fall within the scope of our power analysis without ChatGPT involvement (P-NC vs NP-NC)

Question	Odds Ratio	p-value
Q1: Websites	51.429	<0.001 *
Q2: Recursive	129.167	<0.001 *
Q3: IDE	67.500	<0.001 *
Q7: Languages	∞	<0.001 *
Q8: Compiler	160.000	<0.001 *
Q9: Array	∞	<0.001 *
Q11: Queue	62.000	<0.001 *
Q13: Tree	28.364	<0.001 *
Q19: Detect - FunctionName	37.333	<0.001 *
Q20: Detect - Loop	76.800	<0.001 *

Table 2. Results of Fisher's exact test for screening questions that fall within the scope of our power analysis with ChatGPT involvement (P-NC vs NP-C)

Question	Odds Ratio	p-value
Q1: Websites	37.500	<0.001 *
Q2: Recursive	50.000	<0.001 *
Q8: Compiler	66.667	<0.001 *
Q11: Queue	44.571	<0.001 *
Q19: Detect - FunctionName	54.000	<0.001 *
Q20: Detect - Loop	27.600	<0.001 *

3 CHATGPT EXPERIENCE AND USAGE

Table 3. Number of ChatGPT prompts sent by participants before participating in our study

Number of prompts sent	P-NC	P-C	NP-NC	NP-C
50+	42.3%	76.0%	2.7%	21.2%
19 - 49	15.4%	4.0%	8.1%	9.1%
5 - 18	15.4%	8.0%	5.4%	36.4%
1 - 4	11.5%	4.0%	13.5%	18.2%
0	15.4%	8.0%	70.3%	15.2%

Table 4. Number of participants who used Advanced Prompting to answer the screening questions

Advanced Prompting Used	P-C	NP-C
Yes	44%	15.2%
No	44%	66.7%
Not sure about meaning	12%	18.2%

4 RECOMMENDED SCREENING QUESTIONS BY DANILOVA ET AL.

QD1. Boolean:

Which of these values would be the most fitting for a Boolean?

- True [correct]
- Red
- Solid
- Quadratic
- Small

QD2 Recursive Function:

Choose the answer that best fits the definition of a recursive function.

- A function that calls itself [correct]
- A function that does not have a return value
- A function that can be called from other functions
- A function that runs for an infinite time
- A function that does not require any inputs

QD3 Websites:

Which of these websites is used most frequently by developers as aid for programming?

- Stack Overflow [correct]
- Wikipedia
- Memory Alpha
- LinkedIn
- None of the above

QD4 IDEs known:

Which of these IDEs have you used before?

- Visual Studio [correct]
- Eclipse
- gcc
- HTML
- pyCharm [correct]
- CSS
- jpg
- C++
- UDP
- None of the above

QD5 Array - read index:

Given the array `arr[7, 3, 5, 1, 9]`, what could the command `arr[3]` return? The array starts with 0.

- 1 [correct]
- 7, 3, 5
- 21, 9, 15, 3, 27
- 3
- 10, 6, 8, 4, 12

QD6 Prime check - purpose:

```
1: function func(number):
2:   result = True
3:   for(int i = 2; i<number, i++):
4:     if(number \% i == 0):
5:       result = False
6:       break
7:   return(result)
```

What is the purpose of the pseudo-code algorithm above? Arrays start with 0

- Check if a given number is prime [correct]
- Check if a given number contains the digit 0
- Check if a given number is equal to 0
- Find the smallest digit in the given number
- Check if i is smaller than the given number

5 INTRODUCTION/LANDING PAGE, CONSENT & CAPTCHA

...

- I consent
- I do not consent

[CAPTCHA]

6 STUDY INCLUSION CRITERIA

Q1: How old are you?

Q2: Which programming skill level would you assign yourself to?

- Beginner
- Advanced Beginner
- Intermediate
- Advanced
- Expert
- I have no experience at all

Q3: Do you have a ChatGPT Account?

- yes
- no

Q4: Are you using a laptop or desktop computer to complete this study?

- yes
- no

7 INCENTIVE FOR NON-PROGRAMMERS

In addition to your normal reward for completing the entire survey, you will receive: 0.5 Euro for each of the following correctly solved questions (maximum 14 Euro extra)

8 TASK DESCRIPTION

On the next page you will be presented with programming questions, each with a time limit, one after the other.

(For ChatGPT prompted groups:)

Please use ChatGPT for solving these tasks if possible. ChatGPT is an artificial intelligence to which you can ask questions, and it will provide you with the answers. We are interested in how well ChatGPT can support you with the questions. Please make sure you are logged into ChatGPT and are ready to use it.

(For unprompted groups:)

Please solve the tasks!

9 SCREENING QUESTIONS

Some images have been excluded because of copyright issues and are substituted with brief descriptions. Readers who wish to view these images can contact the authors to request them.

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q1. Websites:

Image description: A blurred screenshot of a Stack Overflow thread is shown.

Which programming-related Webpage do you see?

- Github
- W3schools
- CodeAcedamy
- Visual Studio Code
- Stack Overflow [correct]

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q2. Recursive:

```
function calculate(n):  
    print("Solving ...!")  
    handler = (4 + n * 40) / (n * n)  
    if n <= 1:  
        handler *= 1500  
        handler += 10  
        return 1  
    else:  
        handler += 3000  
        handler += 20  
        if handler > 10000:  
            print("Ressources low!")  
        return n * calculate(n-1)
```

What concept about functions is implemented here?

- Inheritance
- Recursion [correct]
- Activation
- Indention
- Loop

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 15 seconds

When you are ready, click "next" and the countdown starts!

Q3. IDE:

Image description: A full screenshot of the PyCharm (IDE) is shown. This includes the project overview, the debugger and the code editor with a short code snippet.

What type of tool do you see?

- IDE (Integrated Development Environment) [correct]
- VHC (Version Control System)
- PCM (Precise Code Manager)
- CER (Code Error Recognizer)
- ACE (Advanced Code Editor)

Next, a Code Comprehension Question will appear!

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q5. Array Index:

```
a = [1, 2, 3, 4, 5]
print(f(a))
```

```
function f(arr):
    if len(arr) < 4:
        return None

    return arr[3]

    i = 0
    while i < len(arr):
        if arr[i] < 0:
            return None
        i += 1

    j = 0
    while j < 100:
        k = 0
        while k < 100:
            k += 1
        j += 1

    return arr[3]
```

What does this code print? (0-indexed)

- 1
- 7
- 100
- 3
- 0
- 15
- 4 [correct]
- 23

Next, a Code Comprehension Question will appear!

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q6. Prime:

```
function test(number):  
  
    result = True  
  
    if result != True:  
        arr = [3,4,5,3,1,2,3,1,1,1]  
        n = len(arr)  
        for i in range(n):  
            for j in range(0, n-i-1):  
                if arr[j] > arr[j+1] :  
                    arr[j], arr[j+1] = arr[j+1], arr[j]  
                if arr[j] > 0:  
                    print(arr[j])  
                if arr[j] < 10:  
                    print(arr[j] + 1)  
  
    else:  
        for i in range(2, number):  
            if(number % i == 0):  
                result = False  
                break  
    return(result)
```

This code tests if a number ...

- can be inserted into an array
- is equal to 0
- floats
- returns
- is bigger than any $i = 2$
- is a function
- contains the digit 0
- is prime [correct]

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 15 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q7. Languages:

Image description: 15 logos are shown. Five of these logos correspond to known programming languages. The remaining logos are not associated with programming or computer science in general.

Select 2 !!! programming languages (logos)

[correct]:

JavaScript, Python, Java, C and Ruby

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q8. Compiler:

Image description: The image consists of three boxes. There are two arrows. The first arrow points from the first box to the second box and the second arrow points from the second box to the third box. The first box includes a short C code snippet. The second box shows a black box and the third box includes a block of binary code.

What does the black box represent?

- Transformer
- Transcriber
- Compiler [correct]
- Converter
- Binary

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q9. Array:

12	8	7	20	1	values
0	1	2	3	4	indices

Which data structure do you see?

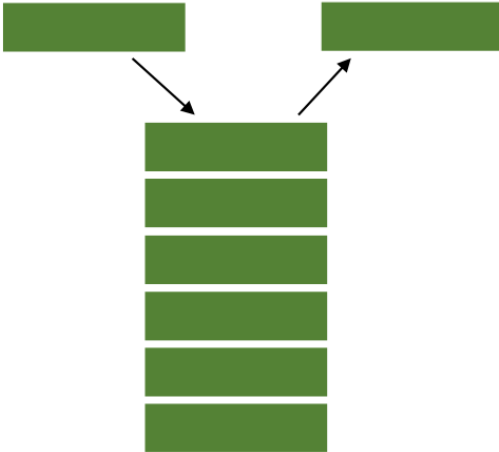
- Counter
- Row
- Sequence
- Array [correct]
- Alignment

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q10. Stack:



Which data structure do you see?

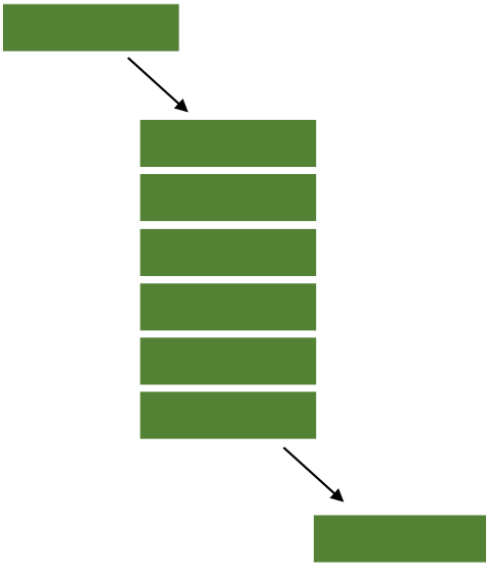
- Builder
- Stack [correct]
- Distributor
- Layer
- Tower

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q11. Queue:



Which data structure do you see?

- Connector
- Queue [correct]
- Pipe
- Processor
- Pass Through

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q12. Linked List:



Which data structure do you see?

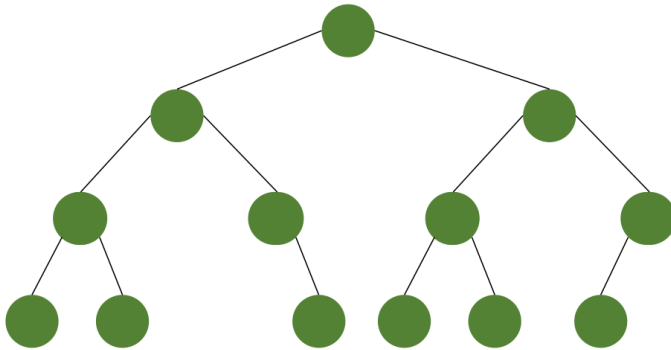
- Expanded Chain
- Connected Header
- Linked List [correct]
- Directed Pointer
- Jumping Extender

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 12 seconds

When you are ready, click "next" and the countdown starts!

Q13. Tree:



Which data structure do you see?

- Tree [correct]
- Root
- Concatinator
- Pendulum
- Branch

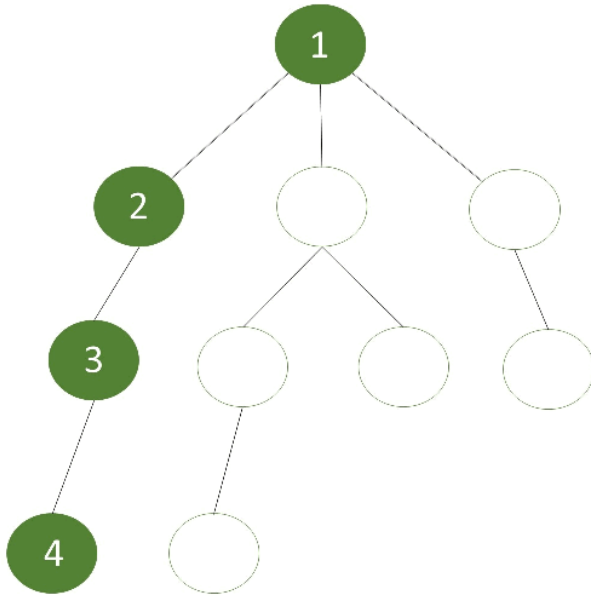
Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 15 seconds
- Response Phase: 15 seconds

When you are ready, click "next" and the countdown starts!

Q14. DFS:

(GIF)



What algorithm do you see?

- Breath-first search
- Vertex-cover search
- Depth-first search [correct]
- Greedy-cover search
- Node-counter search

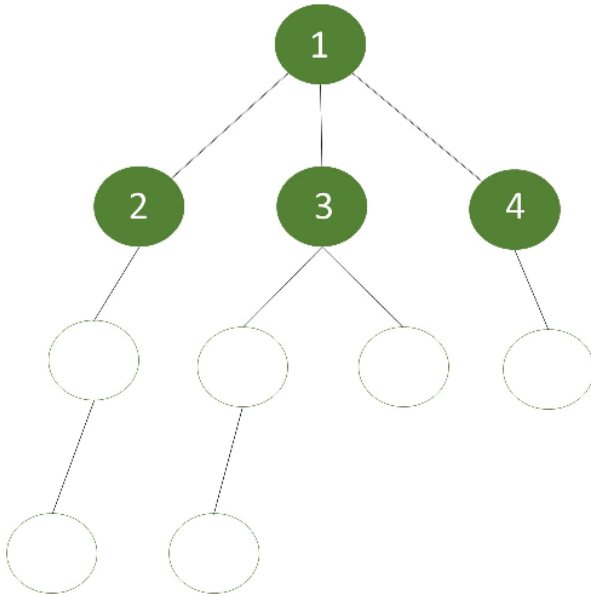
Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 15 seconds
- Response Phase: 15 seconds

When you are ready, click "next" and the countdown starts!

Q15. BFS:

(GIF)



What algorithm do you see?

- Breath-first search [correct]
- Vertex-cover search
- Depth-first search
- Greedy-cover search
- Node-counter search

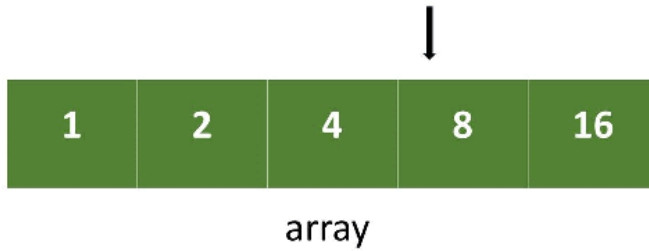
Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 12 seconds
- Response Phase: 15 seconds

When you are ready, click "next" and the countdown starts!

Q16. Run Time:

(GIF)



What is the run time?

- $O(2n)$
- $O(n)$ [correct]
- $O(5n)$
- $O(n^2)$
- $O(n^n)$

Next, a Visual Concept Question will appear! This time 5 questions (5 response phases) about the same visualization follow one after the other without a break!

- Preview Phase (Visualization): 15 econds
- Response Phase 1: 12 seconds
- Response Phase 2: 12 seconds
- Response Phase 3: 12 seconds
- Response Phase 4: 12 seconds
- Response Phase 5: 12 seconds

When you are ready, click "next" and the countdown starts!

Q4. Detect - Bool:

```
/// run function to execute
function run ( calc ):

    str = 2
    b = 1001
    p = 10

    for i in range(0,5):
        calc += str
        b = b * p

    if (calc > 12) == True :
        print("Wow")
```

Select the **boolean value**

<input type="radio"/> <code>for i in range(0,5):</code> <code>calc += str</code> <code>b = b * p</code>	<input type="radio"/> <code>2</code>	<input type="radio"/> <code>run</code>	<input type="radio"/> <code>b</code>
<input type="radio"/> <code>"Wow"</code>	<input type="radio"/> <code>calc</code>	<input type="radio"/> <code>></code>	<input type="radio"/> <code>10</code>
<input type="radio"/> <code>/// run function</code>	<input type="radio"/> <code>True</code>	<input type="radio"/> <code>==</code>	<input type="radio"/> <code>if</code>

[correct]:

<input type="radio"/> <code>True</code>

Q17. Detect - String:

```
/// run function to execute
function run ( calc ):

    str = 2
    b = 1001
    p = 10

    for i in range(0,5):
        calc += str
        b = b * p

    if (calc > 12) == True :
        print("Wow")
```

Select the **string**

<input type="radio"/> if	<input type="radio"/> "Wow"	<input type="radio"/> 10	<input type="radio"/> ==
<input type="radio"/> >	<input type="radio"/> calc	<input type="radio"/> for i in range(0,5): calc += str b = b * p	<input type="radio"/> 2
<input type="radio"/> run	<input type="radio"/> True	<input type="radio"/> /// run function	<input type="radio"/> b

[correct]:

<input type="radio"/> "Wow"

Q18. Detect - Parameter:

```
/// run function to execute
function run ( calc ):

    str = 2
    b = 1001
    p = 10

    for i in range(0,5):
        calc += str
        b = b * p

    if (calc > 12) == True :
        print("Wow")
```

Select the **parameter**

- | | | | |
|---|---|---|---|
| <input type="radio"/> <code>></code> | <input type="radio"/> <pre>for i in range(0,5): calc += str b = b * p</pre> | <input type="radio"/> <code>calc</code> | <input type="radio"/> <code>"Wow"</code> |
| <input type="radio"/> <code>run</code> | <input type="radio"/> <code>10</code> | <input type="radio"/> <code>==</code> | <input type="radio"/> <code>2</code> |
| <input type="radio"/> <code>if</code> | <input type="radio"/> <code>True</code> | <input type="radio"/> <code>b</code> | <input type="radio"/> <code>/// run function</code> |

[correct]:

- `calc`

Q19. Detect - FunctionName:

```
/// run function to execute
function run ( calc ):

    str = 2
    b = 1001
    p = 10

    for i in range(0,5):
        calc += str
        b = b * p

    if (calc > 12) == True :
        print("Wow")
```

Select the **declared function name**

- | | | | |
|--------------------------|--|---------------------------|-----------------------------|
| <input type="radio"/> y | <input type="radio"/> True | <input type="radio"/> run | <input type="radio"/> "Wow" |
| <input type="radio"/> == | <input type="radio"/> for i in range(0,5):
calc += str
b = b * p | <input type="radio"/> 2 | <input type="radio"/> calc |
| <input type="radio"/> b | <input type="radio"/> /// run function | <input type="radio"/> if | <input type="radio"/> 10 |

[correct]:

- run

Q20. Detect - Loop:

```
/// run function to execute
function run ( calc ):

    str = 2
    b = 1001
    p = 10

    for i in range(0,5):
        calc += str
        b = b * p

    if (calc > 12) == True :
        print("Wow")
```

Select the **loop**

- | | | | |
|---|---|---|--|
| <input type="radio"/> <code>b</code> | <input type="radio"/> <code>for i in range(0,5):
calc += str
b = b * p</code> | <input type="radio"/> <code>/// run function</code> | <input type="radio"/> <code>"Wow"</code> |
| <input type="radio"/> <code>True</code> | <input type="radio"/> <code>10</code> | <input type="radio"/> <code>2</code> | <input type="radio"/> <code>calc</code> |
| <input type="radio"/> <code>==</code> | <input type="radio"/> <code>run</code> | <input type="radio"/> <code>></code> | <input type="radio"/> <code>if</code> |

[correct]:

- `for i in range(0,5):
calc += str
b = b * p`

Next, a Code Comprehension Question will appear!

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q21. Conditions:

```
print(test([-2, 3, -5, 9, -9, 8, -13]))
```

```
function test(arr):
    num = arr[3]

    if num % 2 == 0:
        if num > 5:
            if num % 3 == 0:
                return 5
            else:
                return 6
        else:
            return 7

    else:
        if num > 0:
            if num < 10 :
                if num != 8:
                    if (num == 5) and (num == 9):
                        return 1
                    else:
                        return 2
                if num == 3 or num == 9:
                    return 3
            else:
                return 4
```

What does this code print? (0-indexed)

- 4
- 5
- 8
- 3
- 1
- 7
- 6
- 2 [correct]

Next, a Code Comprehension Question will appear!

- Response Phase: 80 seconds

When you are ready, click "next" and the countdown starts!

Q22. Backwards:

```
print(test("scissors"))
```

```
function test(stri):  
    d = ""  
    l = len(stri) - 1  
    while l >= 0:  
        d += stri[l]  
        l -= 1  
    return d
```

What does this code print? (0-indexed)?

- sressocs
- 0
- 4
- srecciss
- 8
- srossics [correct]
- scissor
- srorciss

Next, a Code Comprehension Question will appear!

- Response Phase: 80 seconds

When you are ready, click "next" and the countdown starts!

Q23. Palindrome:

```
print(test("level"))
```

```
function test(s):
    result = 1
    i = 0
    j = len(s) - 1
    while i < j:
        if s[i] != s[j]:
            result -= 1
            return result
        i += 1
        j -= 1
    return result
```

What does this code print? (0-indexed)

- 3
- 5
- 4
- 2
- 0 [correct]
- 3
- 4
- 1

Next, a Code Comprehension Question will appear!

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q24. StringCount:

```
print(test("car123", "card"))
```

```
function test(string1, string2):  
  
    counter = 0  
    i = 0  
    while i < 4:  
        if (string1[i] == string2[i]):  
            counter += 1  
        i += 1  
  
    return counter  
  
    a = 0  
    d = 0  
    length = min(len(string1), len(string2))  
    while a < length:  
        if (string1[a] != string2[a]):  
            d += 1  
        a += 1  
  
    return length - d
```

What does this code print?

- 7
- 4
- 6
- 2
- 8
- 5
- 3 [correct]
- 9

Next, a Code Comprehension Question will appear!

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q25. EndlessLoop:

```
function bubble_sort(array):
    length = len(array)
    i = 0

    while i < length:
        j = 0
        while j < (length - i - 1):
            if array[j] > array[j+1]:
                array[j], array[j+1] = array[j+1], array[j]
            j += 1
            i = i

    if array == [1, 4, 5, 7, 10, 15, 20, 25, 31, 32, 37, 101, 207]:
        return array

    if array == [1, 7, 12, 13, 101, 102, 105, 207, 536, 915]:
        return array

    return array
```

This is an almost working bubble sort function!

What problem for the function does the code in the red box create?

- i is not sorted
- 3 return statements
- no array swaps with i
- no array check of i
- returns i
- returns 0
- endless while loop [correct]
- compares i only once

Next, a Code Comprehension Question will appear!

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q26. Euler Or:

```
print(test([1, 2, 3, 4, 5]))
```

```
function test(a):  
    r = 0  
    i = 0  
  
    while i < len(a):  
        if (a[i] % 3 != 0) or (a[i] % 5 != 0):  
            r += a[i]  
            i += 1  
    return r
```

What does this code print?

- 10
- 6
- 8
- 12
- 4
- 3
- 14
- 15 [correct]

Next, a Code Comprehension Question will appear!

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q27. CrazyCalc:

```
l1 = 2.5  
print(test(l1))
```

```
def test(l):  
    if 4 != l:  
        total = 0  
        i = 3  
        while i < 6:  
            total += i  
            i += 2  
        return total / 10  
    return l
```

What does this code print?

- 2.0
- 0.45
- 1.0
- 2.5
- 1.2
- 2.2
- 0.4
- 0.8 [correct]

Next, a Code Comprehension Question will appear! This time you have to sort code snippets to create a valid required program! (hold the mouse over a code snippet and drag it to the right place).

- Response Phase: 60 seconds

When you are ready, click "next" and the countdown starts!

Q28. Reordering:

Reorder the code snippets to calculate the frequency of „a“ in the string.

```
i = 0
```

```
search = "a"
```

```
count = 0
```

```
i += 1
```

```
while i < length(string):
```

```
count += 1
```

```
if string[i] == search:
```

[correct]:

The order of the first three statements does not matter:

- count = 0
- i = 0
- search = "a"

The remaining four statements must be in that order:

- while i < length(string):
- if string[i] == search:
- count += 1
- i += 1

Next, a Visual Concept Question will appear!

- Preview Phase (Visualization): 7 seconds
- Response Phase: 15 seconds

When you are ready, click "next" and the countdown starts!

Attention Check:

Image description:: A visual representation of a neural net with 6 layers and a single output neuron.

Please just select the answer "machine learning".

- programming
- machine learning [correct]
- code snippets
- turing complete
- algorithm

10 PROGRAMMER-SPECIFIC DEMOGRAPHICS

Q1: Do you work as a software developer in a company?

- yes
- no

Q2: Is software development the main part of your job?

- yes
- no

Q3: How do you rate your proficiency with the following programming, scripting and mark-up languages? Please select all that apply.

Likert scales options: [Extremely proficient, Moderately proficient, Somewhat proficient, Slightly proficient, Not at all proficient, I don't know]

Statements: [JavaScript, Python, Java, Bash/Shell/PowerShell, C#, PHP, TypeScript, C, C++, Rust, Go, Kotlin, Ruby, Assembly]

Q4: How do you rate your proficiency in the following development areas?

Likert scales options: [Extremely proficient, Moderately proficient, Somewhat proficient, Slightly proficient, Not at all proficient, I don't know]

Statements: [Backend development, Frontend development, Full-stack development, Desktop or Enterprise application, development, Embedded application or device, development, Game or graphics development, Mobile development, Quality assurance or testing, Security development, Data science or machine learning, Data or business analysis, UI/UX design, Data engineering, Penetration testing, Security engineering, System architecture,]

Q5: How many years have you been programming in total?

Q6: How many years have you been programming at your job only? (without education, hobby, ...)

Q7: On average, how many hours per week do you work?

Q8: Please select the range that most closely matches the amount of time you typically spend performing software development tasks per week.

- < 5 hours
- 5-10 hours
- 10-20 hours
- 20-30 hours
- 30-40 hours
- > 40 hours

11 NON-PROGRAMMER-SPECIFIC DEMOGRAPHICS

Q1: Do you have any programming skill?

- yes
- no

Q2: How did you acquire your programming skill?

- I don't have any experience
- in a computer science degree or similar field
- in a single programming course
- I am self-taught
- other: (please specify)

Q3: How many years of experience do you have in programming?

12 USAGE AND STRATEGIES OF CHATGPT AND OTHER TOOLS

In this section we want to ask you how you solved the tasks. As a reminder there have been 28 programming questions in total.

Q1: Have you ever heard of ChatGPT before this study?

- yes
- no

Q2: Have you used ChatGPT before this study?

- yes
- no

Q3: How many requests have you made to ChatGPT before this study?

- 0x (none at all)
- 1x - 4x
- 5x - 19
- 19x - 49x
- 50+

Q4: How many questions did you attempt to solve using your own programming skill instead of ChatGPT?

- All questions
- None
- Please specify:

Q5: How many questions did you attempt to solve using your own programming skill?

- All questions
- None
- Please specify:

Q6: Did you attempt to use ChatGPT to solve the tasks in this study?

- Yes
- No

Q7: Do you have a ChatGPT Plus Subscription?

- Yes
- No

Q8: Which other tools did you use to solve the tasks in this study? (Multiple answers allowed)

- Google
- None
- Other(s): (please specify all)

Q9: How many questions did you attempt to solve using ChatGPT?

- All questions
- Please specify:

Q10: Have you stopped using ChatGPT and switched to a different strategy to answer the questions? (Multiple answers allowed)

- No, I used ChatGPT for all questions
- Yes, i switched to using Google
- Yes, i switched to guessing
- Yes, i switched to using my programming knowledge
- Yes, i switched to: (please specify)

Q11: After (approx.) how many questions did you switch your strategy?

Q12: Why did you switch your strategy?

Q13: Briefly describe your approach when you used ChatGPT

Q14: Briefly describe your approach when you used other tools, which are not ChatGPT

Q15: ChatGPT supported me in providing answers to the questions

- Strongly disagree
- Somewhat disagree
- Neither agree nor disagree
- Somewhat agree
- Strongly agree

Q16: Have you used Advanced Prompting when interacting with ChatGPT?

Advanced Prompting refers to techniques such as providing specific instructions, examples, or guiding questions to help ChatGPT better understand your query and provide more accurate responses.

- Yes
- No
- I'm not sure what advanced prompting means

Q17: Did you feel pressure when solving the tasks in this study?

- I did not feel pressured at all
- I felt a little pressured
- I felt very pressured

Q18: Did you feel pressure when solving the Visual Concept Questions?

e.g.,

					112
12	8	7	20	1	value
0	1	2	3	4	indices

Which data structure do you see?

Array
 list
 set
 Class
 Sequence

- I did not feel pressured at all.
- I felt a little pressured.
- I felt somewhat pressured.
- I felt very pressured.
- I felt extremely pressured.

Q19: Did you feel pressure when solving the Code Questions?

e.g.,

0055

What is printed?

```
l1 = 2.5
print(test(l1))
```

```
def test(l):
    if 0 <= l:
        total = 0
        i = 3
        while i <= 0:
            total += 1
            i += 2
        return total / 10
    return l
```

- I did not feel pressured at all.
- I felt a little pressured.
- I felt somewhat pressured.
- I felt very pressured.
- I felt extremely pressured.

13 GENERAL DEMOGRAPHICS

Q1: What describes your current employment status?

- Full-time
- Part-time (19+ hours)
- Unemployed
- Retired
- Self-employed
- Other: (please specify)

Q2: What is your main profession?

Q3: What is your gender?

- Woman
- Man
- Non-binary
- Prefer not to disclose

Prefer to self-describe

Q4: What is your native spoken language?

You can also start typing inside the drop down for a quicker search.

If you can't find your native spoken language, please select 'other'.

[List of spoken languages]

Q5: How comfortable were you taking this survey in English?

- Very comfortable
- Somewhat comfortable
- Neither comfortable nor uncomfortable
- Somewhat uncomfortable
- Not comfortable at all

Q6: Where do you live?

You can also start typing inside the drop down for a quicker search.

[List of countries]

Q7: Where are you currently employed?

- Asia
- Europe
- Africa
- North America
- South America
- Oceania
- I am not permanently employed

Q8: Which of the following best describes the highest level of formal education that you've completed?

- I never completed any formal education
- 10th grade or less (e.g. some American high school credit, German Realschule, British GCSE)
- Secondary school (e.g. American high school, German Realschule or Gymnasium, Spanish or French Baccalaureate, British A-Levels)
- Trade, technical or vocational training
- Some college/university study without earning a degree
- Associate degree (A.A., A.S., etc.)
- Bachelor's degree (B.A., B.S., B.Eng., etc.)
- Master's degree (M.A., M.S., M.Eng., MBA, etc.)
- Professional degree (JD, MD, etc.)
- Other doctoral degrees (Ph.D., Ed.D., etc.)
- Prefer not to disclose
- Other (please specify:)

Q9: Is there anything else you would like to tell us?