**Test: Introduction to Rust**

# Part 1: Multiple Choice

**Circle** the most appropriate answer to each of the following questions.

1. The value **5** in ***Rust*** is a data type called a*(n):*
2. i32 (b) f64 (c) string (d) bool

*[K1]*

1. The expression **‘&’** in ***Rust*** is a data type called a*(n)*:
2. i32 (b) char (c) string (d) f64

*[K1]*

1. The expression **(0,1)** in ***Rust*** is a data type called a*(n)*:
2. integer (b) tuple (c) floating-point (d) Boolean

*[K1]*

1. The value **true** in ***Rust*** is a data type called a*(n)*:
2. bool (b) f64 (c) string (d) i32

*[K1]*

1. What is the output of the following ***Rust*** code.

let a = 5;  
let b = 25;  
let c = a + 10;  
**println!**( “{}”, c + b - a );

(a) -35 (b) 10 (c) 35 (d) 20

*[A1]*

1. Which ***Rust*** code will produce the **integer** value 3?
2. **println!**( “{}”, 15.0 / 5.0 );
3. **println!**( “{}”, 15 % 5 );
4. **println!**( “{}”, 16 // 5 );
5. **println!**( “{}”, 16 / 5 );

*[K1]*

1. Which ***Rust*** code will produce the **floating point** value 3.0 ?
2. **println!**( 3 );
3. **println!**( 15 / 5 );
4. **println!**( 15.0 / 5.0 );
5. **println!**( 15 % 5 );

*[K1]*

1. The output of a ***Rust*** program is:

Welcome to our 2023 computer science class!

*[K1]*

Which code will produce this output?

|  |  |
| --- | --- |
| (a) | **println!**( Welcome to our 2023 computer science class! ); |
| (b) | let year = 2023; **println!**( "Welcome to our {} computer science class!", year ); |
| (c) | let year = 2023; **println!**( "Welcome to our [year] computer science class!"); |
| (d) | let year = 2023; **println!**( Welcome to our, year, computer science class! ); |

1. The output of a program is:

4:58

What ***Rust*** code would produce this output?

|  |  |
| --- | --- |
| (a) | **println!**( 4:58 ); |
| (b) | let minutes = 4; let seconds = 58; **println!**(“{minutes}:{seconds}”); |
| (c) | let minutes = 4; let seconds = 58; **println!**( “{}:{}”, minutes, seconds ); |
| (d) | let minutes = 4; let seconds = 58; **println!**( “minutes:seconds” ); |

*[K1]*

1. The output of a program is:

658

Which ***Rust*** code would produce this value?

|  |  |
| --- | --- |
| (a) | let value = 865; **println!**( “{}”, value / 100 ); |
| (b) | let value = 865; **println!**( “{}”, value % 100 ); |
| (c) | let value = 865; **println!**( “{}{}”, value % 100, value / 100 ); |
| (d) | let value = 865; **println!**(“{}{}”, value / 100, value % 100); |

*[K1]*

1. Select the ***Rust*** code that will produce the following output: **YES**

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | let answer = 42; **if**answer % 2 == 0 {     **println!**("YES"); **} else {**     **println!**("NO");  **}** | (b) | let answer = 42; **if**answer % 2 == 0 {     **println!**("NO"); **} else {**     **println!**("YES");  **}** |
| (c) | let answer = 42; **if**answer % 2 != 0 {     **println!**("YES"); **} else {**     **println!**("NO");  } | (d) | let answer = 42; **if**answer % 2 > 0 {     **println!**("YES"); **} else {**     **println!**("NO");  } |

*[A1]*

1. Select to ***Rust*** code that will produce the following output: **-1**

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | let answer = -15; **if**answer >  0 {     **println!**(“{}”,1); **} else if** answer < 0 **{**     **println!**( “{}”,-1); **} else {**     **println!**( “{}”,0);  } | (b) | let answer = 15; **if**answer >  0 {     **println!**( “{}”,1); **} else if**answer < 0 **{**     **println!**( “{}”,-1); **} else {**     **println!**( “{}”,0);  } |
| (c) | let answer = 0; **if**answer >  0 {     **println!**( “{}”,1); **} else if**answer < 0 {     **println!**( “{}”,-1); **} else {**     **println!**( “{}”,0);  } | (d) | let answer = 15; **if**answer >  0 {     **println!**( “{}”,1); **} else {**     **println!**( “{}”,-1);  } |

*[A1]*

1. Select the ***Rust*** code that will produce the following output: **1 2 3 4 5 6 7 8**

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | **for i** in 1..9 {     **print!**(i);  } | (b) | **for i** in 0..9 **{     print!(**"{} ", i**);**  **}** |
| (c) | **for i** in 1..8 {     **print!**(i);  } | (d) | **for i** in 1..8 {     **print!**("{} ", i);  } |

*[K1]*

1. Select the ***Rust*** code that will produce the following output: **12 10 8 6 4 2 0**

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | **for i** in (0..12).rev() {     **print!**("{} ",i)  } | (b) | **for i** in (0..12).step\_by(2) {     **print!**("{} ",i)  } |
| (c) | **for i** in (0..12).step\_by(2).rev() {     **print!**("{} ",i)  } | (d) | **for i** in (0..13).step\_by(2).rev() {     **print!**("{} ",i)  } |

*[K1]*

1. Select the ***Rust*** code that will produce the following output with **no compiler warnings**:

Happy Break!  
Happy Break!  
Happy Break!

Happy Break!

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | **for** i in 0..4 {     **println!**("Happy Break!");  } | (b) | **for \_**i in 1..4 {     **println!**(" Happy Break!");  } |
| (c) | **for \_**i in 0..4 {  **println!**(" Happy Break!");  } | (d) | **for**i in 1..4 {     **println!**(" Happy Break!"); } |

*[K1]*

1. Select the ***Rust*** code that will output the **sum** of the **first 9 whole numbers**, producing the following output:

**45**

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | let mut total = 0; **for** \_i in 0..9 {     total++;  } **println!**(“{}”,total); | (b) | let mut total = 0; **for**i in 0..10 {     total += i;  } **println!**( “{}”,total); |
| (c) | let total = 0; **for**i in 0..10 {     total = total + i;  }  **println!**( “{}”,total); | (d) | let mut total = 1; **for**\_i in 0..10 {     total += 1;  } **println!**( “{}”,total); |

*[A1]*

1. Select the following ***Rust*** code that will produce the output:

**85**

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | let marks = [65, 85, 77, 90]; **println!**(  “{:?}”, marks ); | (b) | let marks = [65, 85, 77, 90]; **println!**(  “{}”, marks[0] ); |
| (c) | let marks = [65, 85, 77, 90]; **println!**(  “{}”, marks[1] ); | (d) | let marks = [65, 85, 77, 90]; **println!**(  “{}”, marks[2] ); |

*[K1]*

1. A **function** in ***Rust*** is defined as follows

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | fn triple(x: i32) -> i32 {  x\*3;  } | (b) | fn triple(x) {  x\*3  } |
| (c) | fn triple(x: i32) -> i32 {  x\*3  } | (d) | func triple(int x) int {  x\*3  } |

*[K1]*

1. Select the ***Rust*** code that will output the value of 4! *(4x3x2x1)*, producing the following output:

24

|  |  |  |  |
| --- | --- | --- | --- |
| (a) | let product = 0; **for**i **in** 1..5 {     product = product\* i;  } **println!**(“{}”, product); | (b) | let product = 1; **for**i **in** 0..4 {     product = product\* i;  } **println!**( “{}”, product); |
| (c) | let mut product = 1; **for**i **in** 0..5 {     product = product\* i;  } **println!**( “{}”, product); | (d) | let mut product = 1; **for**i **in** 1..5 {     product = product\* i;  } **println!**( “{}”, product); |

*[A1]*

1. What symbol do you use to make a comment in ***Rust***?
2. # (b) @ (c) <> (d) //

*[K1]*

# Part 2: Program Analysis

Answer the following questions about the given ***Rust*** source code using as detailed answers as possible.

# Program A

Given the following ***Rust*** source code,

A screenshot of a computer program

Description automatically generated

1. What **type of data** is stored in the array called **values**?

### *[T1]*

1. The Rust compiler generates an **error** on **line 12**. What needs to be to be done to fix the error?

### *[T1]*

1. After fixing the error on line 12, there is another error on **line 16**. **What** is the error and **how** do you fix it?

### *[T2]*

1. Briefly **describe** what this program does when you run it without errors.

### *[T1]*

1. State the **output** of the final program.

### *[T2]*

# Program B

Given the following ***Rust*** source code,

A screenshot of a computer program

Description automatically generated

1. What **type of data** is stored in the array called **dna**?

### *[T1]*

1. What **type of data** is returned from the function **info**?

### *[T1]*

1. Briefly **describe** what this program does when you run it.

### *[T2]*

1. Write the exact output of the program.

### *[T1]*