Informatik I - Exercise Session
Past Exam Questions

[Exam 2022-08 MAVT] Expression Evaluation

Remark to Type and Value Questions: The keyword `auto` means that the type of the expression is determined by the compiler. In the following it thus stands for the expression type that you need to identify.

1. Provide type and value of variable c.

```
1 int a = 5;
2 int b = 1;
3 auto c = (9 * a + b) % a;
```

2. Provide type and value of variable c.

```
1 int a = 5;
2 double b = 1;
3 auto c = (9.0 * a + b) / a;
```

[Exam 2022-08 MAVT] Expression Evaluation

Remark to Type and Value Questions: The keyword `auto` means that the type of the expression is determined by the compiler. In the following it thus stands for the expression type that you need to identify.

1. Provide type and value of variable c.

```
1 int a = 5;
2 int b = 1;
3 auto c = (9 * a + b) % a;
```

`int`

2. Provide type and value of variable c.

```
1 int a = 5;
2 double b = 1;
3 auto c = (9.0 * a + b) / a;
```

[Exam 2022-08 MAVT] Expression Evaluation

Remark to Type and Value Questions: The keyword `auto` means that the type of the expression is determined by the compiler. In the following it thus stands for the expression type that you need to identify.

1. Provide type and value of variable c.

```
1 int a = 5;
2 int b = 1;
3 auto c = (9 * a + b) % a;
```

`int , 1`

2. Provide type and value of variable c.

```
1 int a = 5;
2 double b = 1;
3 auto c = (9.0 * a + b) / a;
```

[Exam 2022-08 MAVT] Expression Evaluation

Remark to Type and Value Questions: The keyword `auto` means that the type of the expression is determined by the compiler. In the following it thus stands for the expression type that you need to identify.

1. Provide type and value of variable c.

```
1 int a = 5;
2 int b = 1;
3 auto c = (9 * a + b) % a;
```

`int , 1`

2. Provide type and value of variable c.

```
1 int a = 5;
2 double b = 1;
3 auto c = (9.0 * a + b) / a;
```

`double`

[Exam 2022-08 MAVT] Expression Evaluation

Remark to Type and Value Questions: The keyword `auto` means that the type of the expression is determined by the compiler. In the following it thus stands for the expression type that you need to identify.

1. Provide type and value of variable c.

```
1 int a = 5;
2 int b = 1;
3 auto c = (9 * a + b) % a;
```

`int , 1`

2. Provide type and value of variable c.

```
1 int a = 5;
2 double b = 1;
3 auto c = (9.0 * a + b) / a;
```

`double , 9.2`

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system F*.

$$F^*(\beta = 2, p = 3, e_{\text{min}} = -1, e_{\text{max}} = 4)$$

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- 1.25 can be represented exactly in the floating point system $F*$.
- There is no number $Z \in F^*$ such that $0.0625 < Z < 0.25$.
- 3.25 can be represented exactly in the floating point system $F*$.

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system F*.

$$F^*(\beta = 2, p = 3, e_{\text{min}} = -1, e_{\text{max}} = 4)$$

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- 1.25 can be represented exactly in the floating point system $F*$.
  TRUE
- There is no number $Z \in F^*$ such that $0.0625 < Z < 0.25$.
- 3.25 can be represented exactly in the floating point system $F*$.

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system F*.

$$F^*(\beta = 2, p = 3, e_{\min} = -1, e_{\max} = 4)$$

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- 1.25 can be represented exactly in the floating point system $F*$.
  TRUE , $1.01 * 2^0$
- There is no number $Z \in F^*$ such that $0.0625 < Z < 0.25$.
- 3.25 can be represented exactly in the floating point system $F*$.

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system F*.

$$F^*(\beta = 2, p = 3, e_{\min} = -1, e_{\max} = 4)$$

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- 1.25 can be represented exactly in the floating point system $F*$.
  TRUE , $1.01 * 2^0$
- There is no number $Z \in F^*$ such that $0.0625 < Z < 0.25$.
  TRUE
- 3.25 can be represented exactly in the floating point system $F*$.

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system F*.

$$F^*(\beta = 2, p = 3, e_{\min} = -1, e_{\max} = 4)$$

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- 1.25 can be represented exactly in the floating point system $F*$.
  TRUE , $1.01 * 2^0$
- There is no number $Z \in F^*$ such that $0.0625 < Z < 0.25$.
  TRUE , the smallest number that can be represented is 0.5 (i.e., $1.0 * 2^{-1}$)
- 3.25 can be represented exactly in the floating point system $F*$.

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system F*.

$$F^*(\beta = 2, p = 3, e_{\min} = -1, e_{\max} = 4)$$

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- 1.25 can be represented exactly in the floating point system $F*$.
  TRUE , $1.01 * 2^0$
- There is no number $Z \in F^*$ such that $0.0625 < Z < 0.25$.
  TRUE , the smallest number that can be represented is 0.5 (i.e., $1.0 * 2^{-1}$)
- 3.25 can be represented exactly in the floating point system $F*$.
  FALSE

[Exam 2022-08 MAVT] Normalized Floating Point Systems

Answer the following questions regarding the normalized floating point system F*.

$$F^*(\beta = 2, p = 3, e_{\min} = -1, e_{\max} = 4)$$

Reminder: For F*, the precision (number of digits) includes the leading bit.

True or false?

- 1.25 can be represented exactly in the floating point system $F*$.
  TRUE , $1.01 * 2^0$
- There is no number $Z \in F^*$ such that $0.0625 < Z < 0.25$.
  TRUE , the smallest number that can be represented is 0.5 (i.e., $1.0 * 2^{-1}$)
- 3.25 can be represented exactly in the floating point system $F*$.
  FALSE , $3.25 = 1.101 * 2^1$ would require precision $p \geq 4$

[Exam 2022-08 252-08(47/48/56)] Loop Termination

```
1 int sum = 17;
2 int i = 1;
3
4 do {
5   i += sum;
6   sum = sum / 2;
7 } while (i > sum && sum >= 0);
8
9 std::cout << sum;
```

Which statement describes the output best?

- 17
- 8
- Never terminates
- 18

[Exam 2022-08 252-08(47/48/56)] Loop Termination

```
1 int sum = 17;
2 int i = 1;
3
4 do {
5   i += sum;
6   sum = sum / 2;
7 } while (i > sum && sum >= 0);
8
9 std::cout << sum;
```

Which statement describes the output best?

- ○ 17
- ○ 8
- ○ Never terminates CORRECT
- ○ 18

[Exam 2022-08 252-08(47/48/56)] Loop Termination

```
1 int sum = 17;
2 int i = 1;
3
4 do {
5   i += sum;
6   sum = sum / 2;
7 } while (i > sum && sum >= 0);
8
9 std::cout << sum;
```

Which statement describes the output best?

- ○ 17
- ○ 8
- ○ Never terminates CORRECT
- ○ 18

- Division of two positive `int`s cannot be negative.
  ⇒ `sum >= 0` is always `true`

[Exam 2022-08 252-08(47/48/56)] Loop Termination

```
1 int sum = 17;
2 int i = 1;
3
4 do {
5   i += sum;
6   sum = sum / 2;
7 } while (i > sum && sum >= 0);
8
9 std::cout << sum;
```

Which statement describes the output best?

- ○ 17
- ○ 8
- ○ Never terminates CORRECT
- ○ 18

- Division of two positive `int`s cannot be negative. ⇒ `sum >= 0` is always `true`
- After the first execution of the `do` block: `i > sum`.

[Exam 2022-08 252-08(47/48/56)] Loop Termination

```cpp
int sum = 17;
int i = 1;

do {
  i += sum;
  sum = sum / 2;
} while (i > sum && sum >= 0);

std::cout << sum;
```

Which statement describes the output best?

- ○ 17
- ○ 8
- ○ Never terminates CORRECT
- ○ 18

- Division of two positive `int`s cannot be negative. ⇒ `sum >= 0` is always `true`
- After the first execution of the `do` block: `i > sum`. `sum` is monotonically decreasing, `i` is monotonically increasing.

[Exam 2022-08 252-08(47/48/56)] Loop Termination

```cpp
int sum = 17;
int i = 1;

do {
  i += sum;
  sum = sum / 2;
} while (i > sum && sum >= 0);

std::cout << sum;
```

Which statement describes the output best?

- ○ 17
- ○ 8
- ○ Never terminates CORRECT
- ○ 18

- Division of two positive `int`s cannot be negative. ⇒ `sum >= 0` is always `true`
- After the first execution of the `do` block: `i > sum`. `sum` is monotonically decreasing, `i` is monotonically increasing. ⇒ `i > sum` is always `true`