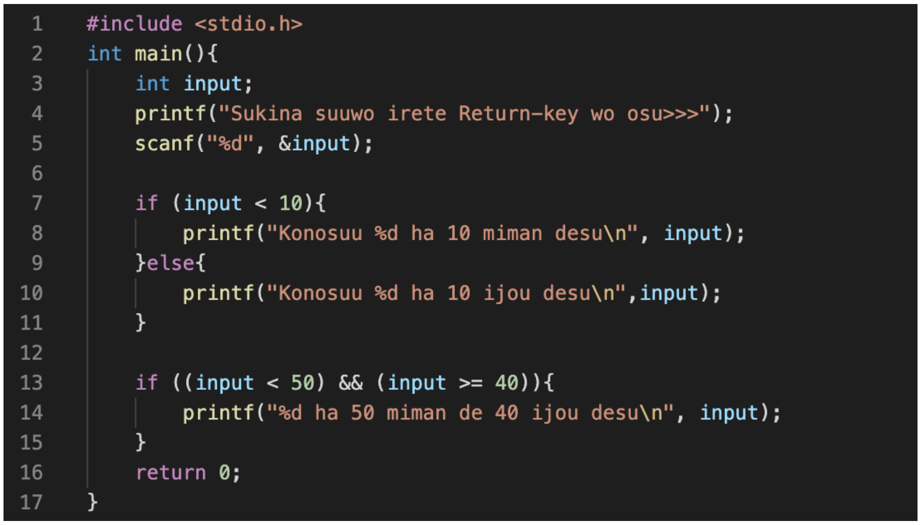
202018021
Fu, Ziyu
Programming I Chapter 2 Task 1, 2, 3
2021/04/29

Task 1

1. Objective
   To understand the sample code provided in the textbook and explore the mechanisms of if/else conditional statements.

2. Strategy of solving
   - Inspect the structure of the provided sample code
   - Run the code
   - Observe the output

3. Program code



```
1   #include <stdio.h>
2   int main(){
3       int input;
4       printf("Sukina suuwo irete Return-key wo osu>>>");
5       scanf("%d", &input);
6
7       if (input < 10){
8           printf("Konosuu %d ha 10 miman desu\n", input);
9       }else{
10          printf("Konosuu %d ha 10 ijou desu\n",input);
11      }
12
13      if ((input < 50) && (input >= 40)){
14          printf("%d ha 50 miman de 40 ijou desu\n", input);
15      }
16      return 0;
17  }
```

4. Results and discussions
   Console output:

```
Sukina suuwo irete Return-key wo osu>>>50
Konosuu 50 ha 10 ijou desu

Sukina suuwo irete Return-key wo osu>>>40
Konosuu 40 ha 10 ijou desu
40 ha 50 miman de 40 ijou desu

Sukina suuwo irete Return-key wo osu>>>20
Konosuu 20 ha 10 ijou desu

Sukina suuwo irete Return-key wo osu>>>6
Konosuu 6 ha 10 miman desu
```

This program consists of two `if` statements, on `line 7` and `line 13`. Because the `if` statement on `line 7` is paired with an alternative `else`, the logic is bound to fall into either one of the situations. On the other hand, the `if` statement on `line 13` can potentially be skipped all together, if the condition is not fulfilled.

When `input = 50`, for the first `if-else` statement, it falls under the `else` condition, hence skipping the command on `line 8` and executing the command on `line 10`; for the second `if` statement, it doesn't fulfill the condition, hence skipping the command on `line 14`.

When `input = 40`, for the first `if-else` statement, it falls under the `else` condition, hence skipping the command on `line 8` and executing the command on `line 10`; for the second `if` statement, it fulfills the condition, so it also executes the command on `line 14`.

When `input = 20`, for the first `if-else` statement, it falls under the `else` condition, hence skipping the command on `line 8` and executing the command on `line 10`; for the second `if` statement, it doesn't fulfill the condition, hence skipping the command on `line 14`.

When `input = 6`, for the first `if-else` statement, it falls under the first `if` condition, hence executing the command on `line 8` and skipping the command on `line 10`; for the second `if` statement, it doesn't fulfill the condition, hence skipping the command on `line 14`.

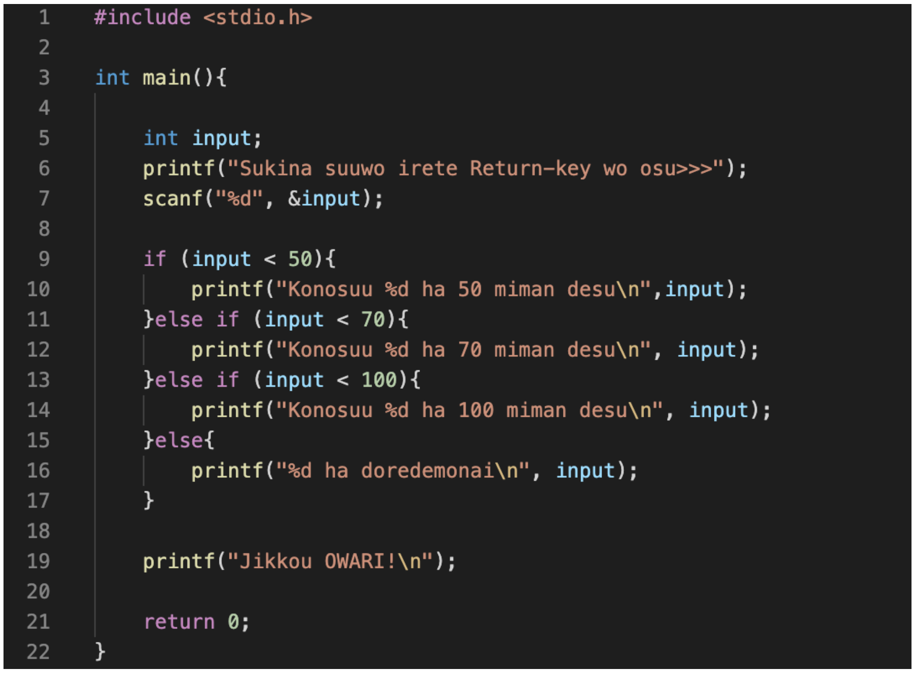
Task 2

1. Objective

   To understand the sample code provided in the textbook and explore the mechanisms of if/else conditional statements.

2. Strategy of solving
   - Inspect the structure of the provided sample code
   - Run the code
   - Observe the output

3. Program code



```c
#include <stdio.h>

int main(){

    int input;
    printf("Sukina suuwo irete Return-key wo osu>>>");
    scanf("%d", &input);

    if (input < 50){
        printf("Konosuu %d ha 50 miman desu\n",input);
    }else if (input < 70){
        printf("Konosuu %d ha 70 miman desu\n", input);
    }else if (input < 100){
        printf("Konosuu %d ha 100 miman desu\n", input);
    }else{
        printf("%d ha doredemonai\n", input);
    }

    printf("Jikkou OWARI!\n");

    return 0;
}
```

4. Results and discussions

   Console output:

```
Sukina suuwo irete Return-key wo osu>>>101
101 ha doredemonai
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>80
Konosuu 80 ha 100 miman desu
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>70
Konosuu 70 ha 100 miman desu
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>50
Konosuu 50 ha 70 miman desu
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>10
Konosuu 10 ha 50 miman desu
Jikkou OWARI!
```

This program consists of one single conditional statement with a nested **`else-if`** ladder. The test-expressions are evaluated from top to bottom. Whenever a **`true`** test-expression is found, the logic will fall into that condition and execute the commands inside, and it would break out of the entire chain without checking the conditions after. This ladder also has an **`else`** alternative, so if none of the test conditions are **`true`**, the program will execute the default commands. Because all of these conditionals are chained together with **`else-if`** statements, only one of the situations will be executed.

The **`printf`** command on **`line 19`**, however, lies outside the conditions, hence it will always be executed.

When **`input = 101`**, it satisfies none of the **`if`** or **`else-if`** conditions, hence the program falls into the **`else`** alternative, executing the command on **`line 16`**.

When **`input = 80`**, it satisfies the **`else-if`** condition on **`line 13`**, hence executing the command on **`line 14`**.

When **`input = 70`**, it satisfies the **`else-if`** condition on **`line 13`**, hence executing the command on **`line 14`**.

When **`input = 50`**, it satisfies the **`else-if`** condition on **`line 11`**, hence executing the command on **`line 12`**.

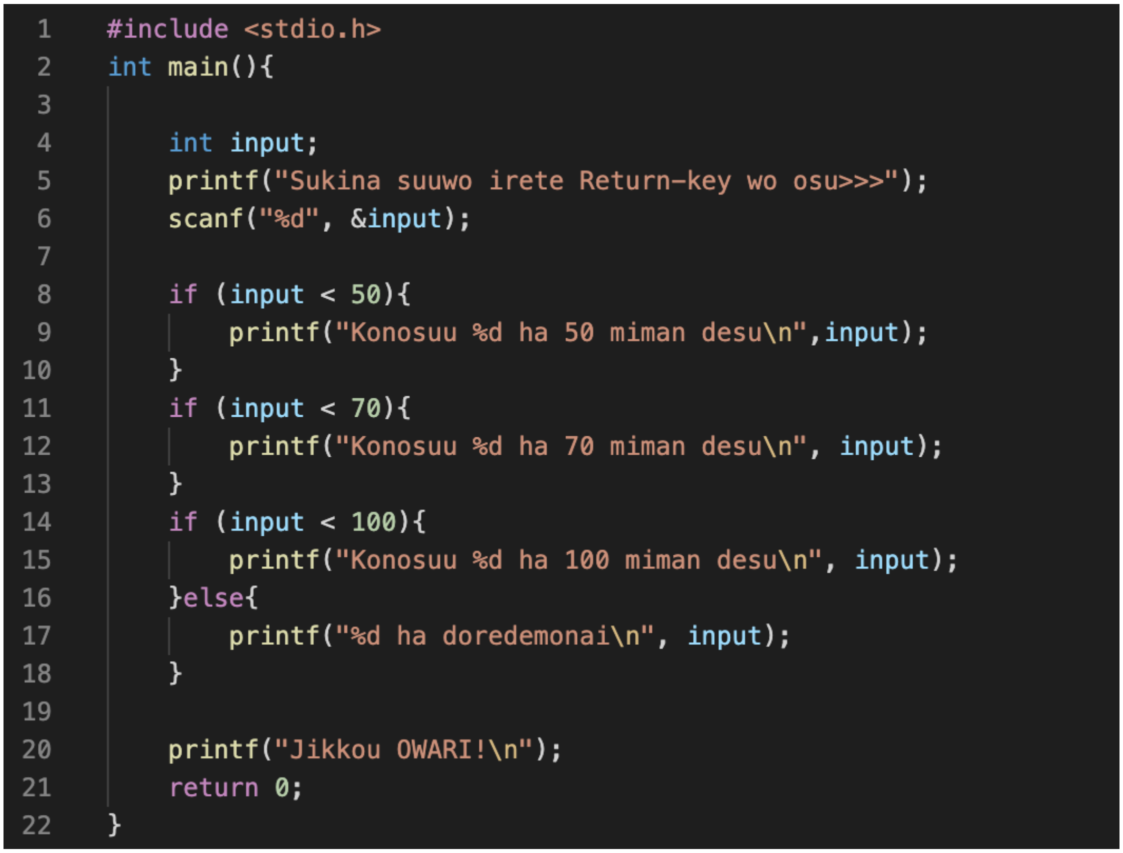
When **`input = 10`**, it satisfies the **`else-if`** condition on **`line 19`**, hence executing the command on **`line 10`**.

Task 3

1. Objective
   To understand the sample code provided in the textbook and explore the mechanisms of if/else conditional statements.

2. Strategy of solving
   - Inspect the structure of the provided sample code
   - Run the code
   - Observe the output

3. Program code



```c
#include <stdio.h>
int main(){

    int input;
    printf("Sukina suuwo irete Return-key wo osu>>>");
    scanf("%d", &input);

    if (input < 50){
        printf("Konosuu %d ha 50 miman desu\n",input);
    }
    if (input < 70){
        printf("Konosuu %d ha 70 miman desu\n", input);
    }
    if (input < 100){
        printf("Konosuu %d ha 100 miman desu\n", input);
    }else{
        printf("%d ha doredemonai\n", input);
    }

    printf("Jikkou OWARI!\n");
    return 0;
}
```

4. Results and discussions
   Console output:

```
Sukina suuwo irete Return-key wo osu>>>101
101 ha doredemonai
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>80
Konosuu 80 ha 100 miman desu
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>70
Konosuu 70 ha 100 miman desu
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>50
Konosuu 50 ha 70 miman desu
Konosuu 50 ha 100 miman desu
Jikkou OWARI!

Sukina suuwo irete Return-key wo osu>>>10
Konosuu 10 ha 50 miman desu
Konosuu 10 ha 70 miman desu
Konosuu 10 ha 100 miman desu
Jikkou OWARI!
```

This program has the same testing conditionals as the program from Task 2, however, the `if` statements are set up in a very different way. Instead of a long chain of `else-if` ladder, this program consists of 3 separate `if` statements. The `if` statements on `line 8` and `line 11` don't have `else` alternatives, making it possible to either satisfy or skip the logic altogether. The `if` statement on `line 14` is paired with an `else` alternative, so the logic is bound to fall into either one of these situations. This setup is very different from the `else-if` setup in Task 2 because it is possible to satisfy all three conditions and have them all executed in one run, as opposed to the setup in Task 2 where only one situation can be fulfilled.

Similarly, the `printf` command on `line 20`, again lies outside the conditions, hence it will always be executed.

When `input = 101`, it satisfies none of the `if` conditions, hence the program falls into the `else` alternative for the last `if` condition on line 14, executing the command on `line 17`.

When `input = 80`, it satisfies one `if` condition on `line 14`, hence executing the command on `line 15`.

When `input = 70`, it satisfies one `if` condition on `line 14`, hence executing the command on `line 15`.

When `input = 50`, it satisfies two `if` conditions on `line 11` and `line 14`, hence executing the commands on `line 12` and `line 15.`

When `input = 10`, it satisfies all three `if` conditions on `line 8`, `line 11` and `line 14`, hence executing the commands on `line 9`, `line 12` and `line 15.`