L05 Tasks

L05-T1: Three different simple versions of a function

L05-T2: Function with parameters, value comparison

L05-T3: Function with parameters, string check

L05-T4: Compressed string

L05-T5: Menu-based program / Calculator

Submission:

- Submit L05-T4 and L05-T5 to CodeGrade via Moodle before the deadline.

Note:

- Be especially careful with spaces so that your output follows the sample output. Note that also each input-string in the program is ending with '\n'. The reason for this is that it makes the output in CodeGrade more readable.

- If you see some "Code Structure Tests" hidden in CodeGrade but they are not mentioned in the task description, you don't need to worry about them. They are just there to make sure the code is ok.

L05-T1: Three different simple versions of a function

In this task, you need to write three simple functions.

a) Print text inside a function

Create a function named print_lines() that just prints the text “This is printed inside the function”.

Note that this function is not typical, because it:

1. Does not receive any parameters.
2. Does not return any values.

b) Input parameters and calculation

Write a function `process_number(nr)` that:

1. Receives an integer nr as a parameter.
2. Prints the received value.
3. Calculates the square of the integer and prints it.
4. Does not return any values.

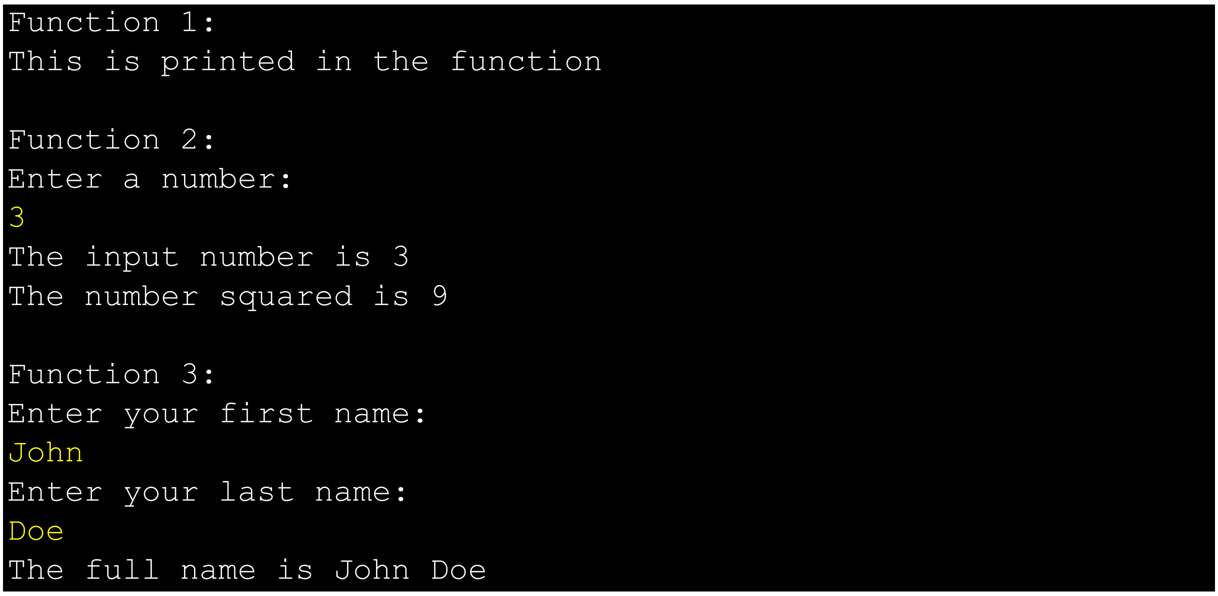
Describe what happens when you execute the following lines of code (this will be presented in the exercise class)

```
number = 5

process_number(number)

print(number)
```

c) Printing the name

Write a function print_whole_name(), which prompts the user to enter their last and first name, it then combines both the first name and last name into a full name, and then returns the full name to the main program.

Example run 1:



```
Function 1:
This is printed in the function

Function 2:
Enter a number:
3
The input number is 3
The number squared is 9

Function 3:
Enter your first name:
John
Enter your last name:
Doe
The full name is John Doe
```

L05-T2: Function with parameters, value comparison

Write a program that asks the user for two integers. Add a function to your program that receives the integers requested from the user as parameters and tests which of the given numbers is larger. Your function should return the larger value or, in case of a tie, the first of the given numbers, and print the message "The numbers are the same size.".

Your program will then ask for a third number, which will be subtracted from the return value returned by the function. Finally, the program uses the same function to test which is larger, the original smaller value, or the return value after subtraction.

Example run 1:

```
Enter the first integer:
42
Enter the second integer:
39
42 is greater than 39
Enter the integer that is subtracted from the larger:
3
The integers are the same.
```

Example run 2:

```
Enter the first integer:
55
Enter the second integer:
55
The integers are the same.
Enter the integer that is subtracted from the larger:
-5
60 is greater than 55
```

L05-T3: Function with parameters, string check

Write a function that takes two strings as parameters and returns the Boolean value True if the first string is entirely contained within the second, and False otherwise. The solution should be based on nested loops, **the use of the operator `in` is not allowed in testing**.

Write also a main program that prompts the user to enter two strings and then calls this function to determine whether the first string can be found within the second string.

Example run 1:

```
Enter the first string:
part
Enter the second string:
apartment
The first string can be found in the second string.
```

Example run 2:

```
Enter the first string:
tea
Enter the second string:
treat
The first string cannot be found in the second string.
```

L05-T4: Compressed string

(Submit this task to CodeGrade on Moodle)

Write a function that takes a string as input and returns a compressed version of the string. For example, for the input "AAAAAAAAAAAAAHHHEEM" (19 characters), the function should return "A13H3E2M" (8 characters). The idea is that if a letter X appears *N times* in a row, this is compressed as XN. Note that it is "sensible" to add the count only if it's greater than 1, otherwise the text will get longer.

Compute also the compression ratio, that is, the length of compressed text divided by the length of the original text. Round the ratio to two decimals using round() function.

Note: **make sure that you have a function that receive a string as input and returns a string. This will be tested in the "Code structure test" in CodeGrade. If the structure test fails, the remaining tests will not be graded.**

Example run 1:

```
Give a string to compress:
python
Compressed string: python
Compressing ratio 1.0
```

Example run 2:

```
Give a string to compress:
aaaabbbbbbbcdddddddddeekkkkkkk
Compressed string: a4b7cd9e2k7
Compressing ratio 0.37
```

Example run 3:

```
Give a string to compress:
AAAAABDDUDFGGNSSSSNXUWXXX
Compressed string: A5BD2UDFG2NS4NXUWX3
Compressing ratio 0.76
```

L05-T5: Menu-based program / Calculator. Continuing from L04-T5

(Submit this task to CodeGrade on Moodle)

In task L03-T3 we made our first version of a calculator. In task L04-T5, we added the repeating structure. The program could be run many times in a row and stopped by selecting Stop from the menu. This time, the program is divided into several subprograms and the functionality of the program is simplified.

Note: **make sure that your code is having these functions with exact names and parameters. This will be tested in the “Code structure tests” in CodeGrade. If these structure tests fail, the remaining tests will not be graded.**

Divide you program into the following subprograms:

• **menu():** This function prints the menu, asks the user for the desired operation number and returns it to the calling program as an integer.

• **enter_integer(text):** This function receives an instruction to be displayed to the user as a string parameter text and returns the integer given by the user to the calling program.

• **sum(value1, value2):** This function receives two integers as parameters, calculates `value1 + value2` and **returns a string** to be printed to the calling program.

• **subtract(value1, value2):** This function receives two integers as parameters, calculates `value1 - value2`, and **returns a string** to be printed to the calling program.

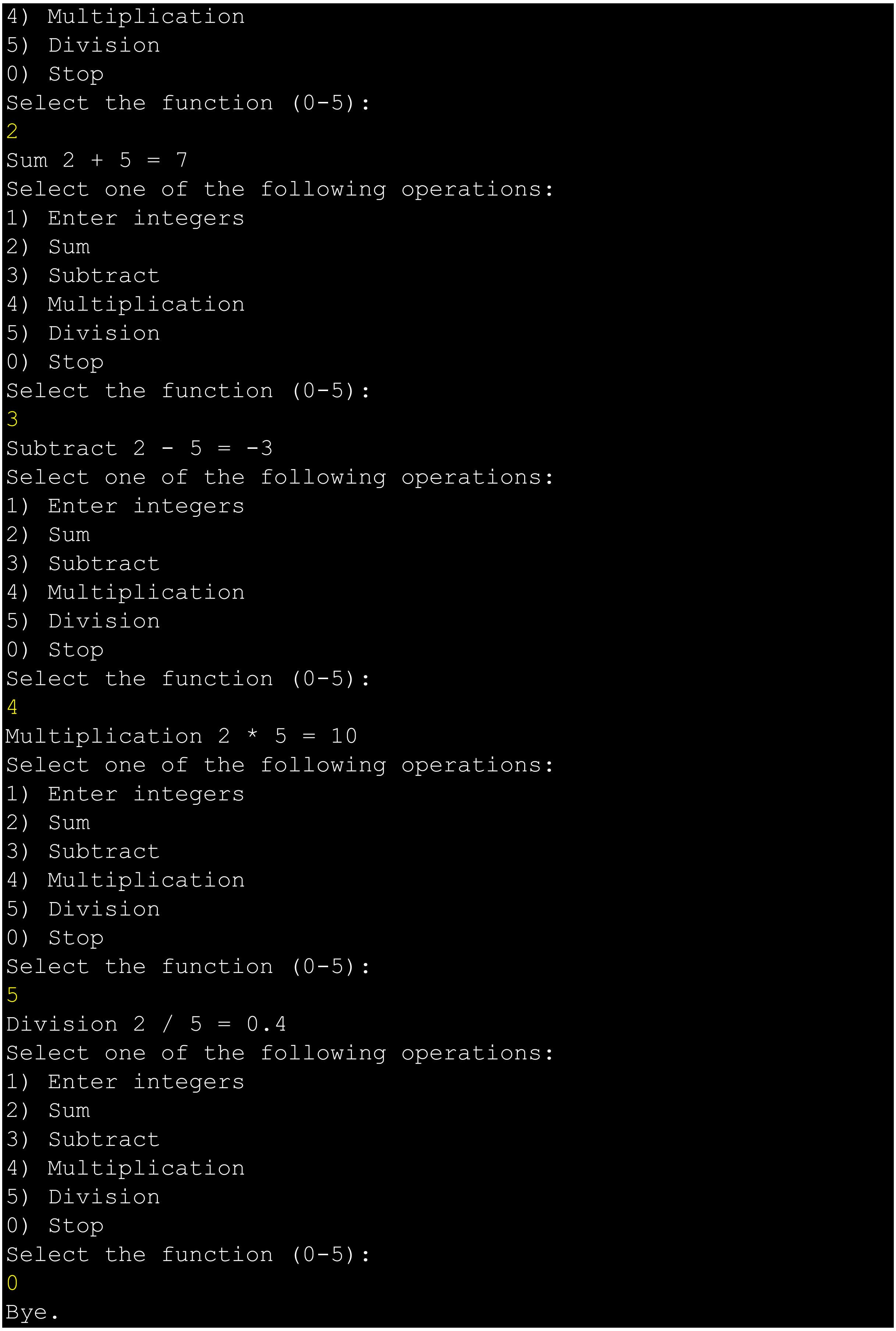
• **multiplication(value1, value2):** This function receives two integers as parameters, calculates `value1 * value2` and **returns a string** to be printed to the calling program.

• **division(value1, value2):** This function receives two integers and calculates `value1 / value2` and **returns a string** to be printed to the calling program. If the divider is 0, the function returns the string "**You cannot divide by zero.**”. Round the result of the division calculation to **two decimal** places with the round function.

• **main()**: The main program contains the repeat and select structures of the previous task, and calls the above subroutines whenever necessary. The main program is therefore responsible for coordinating the entire program based on user inputs.

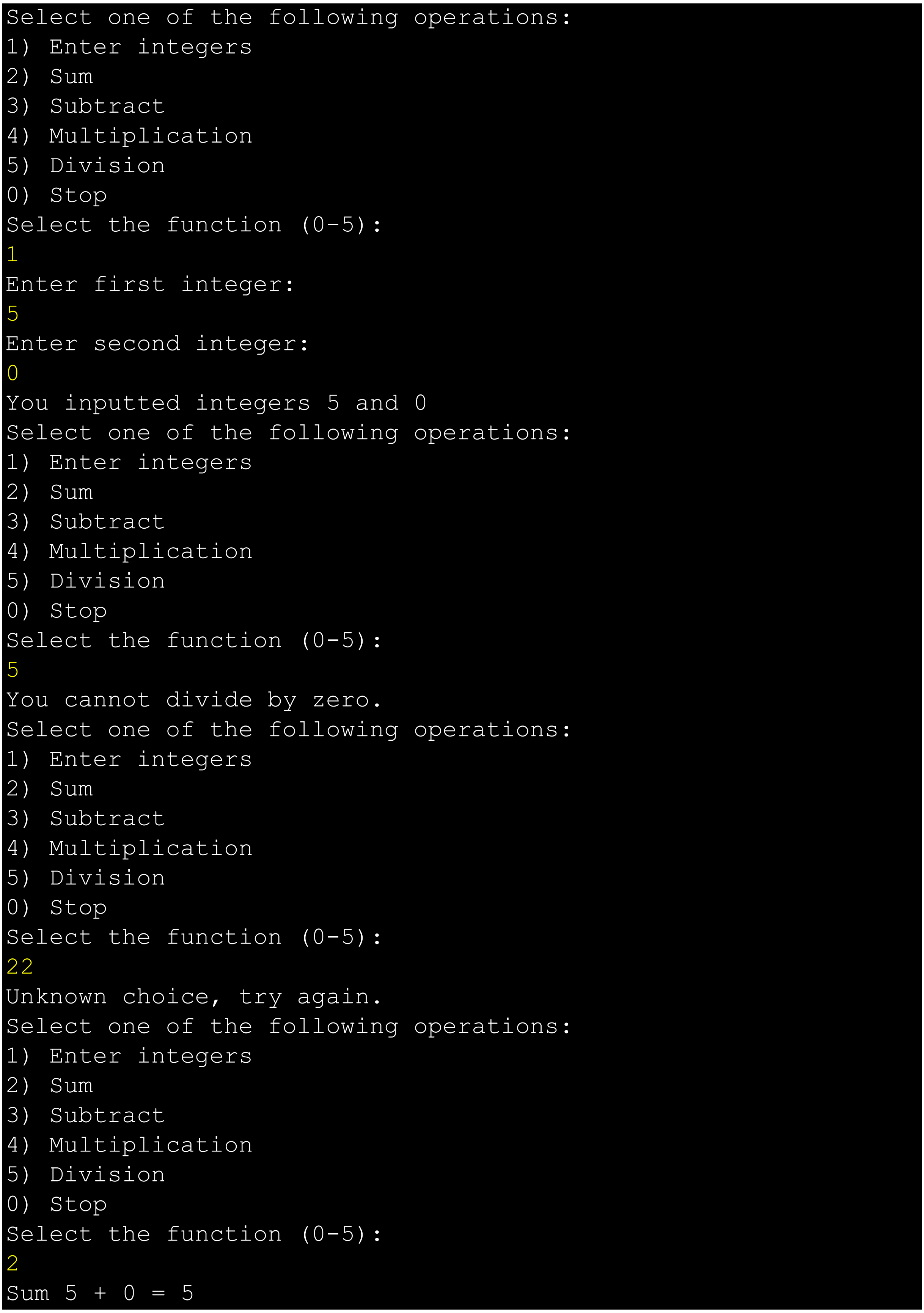
Example run 1:

```
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
1
Enter first integer:
2
Enter second integer:
5
You inputted integers 2 and 5
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
```

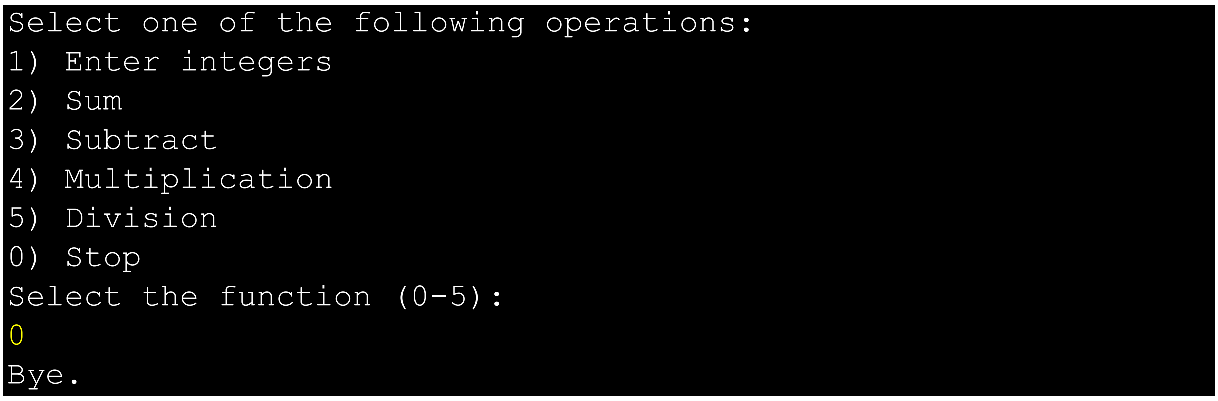


```
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
2
Sum 2 + 5 = 7
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
3
Subtract 2 - 5 = -3
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
4
Multiplication 2 * 5 = 10
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
5
Division 2 / 5 = 0.4
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
0
Bye.
```

Example run 2:



```
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
1
Enter first integer:
5
Enter second integer:
0
You inputted integers 5 and 0
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
5
You cannot divide by zero.
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
22
Unknown choice, try again.
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
2
Sum 5 + 0 = 5
```



```
Select one of the following operations:
1) Enter integers
2) Sum
3) Subtract
4) Multiplication
5) Division
0) Stop
Select the function (0-5):
0
Bye.
```