

ESS111 : Programming 1 (C Programming)

LAB - 4

Due: 16 December, 2020 @ 11:59 pm

Part A (to be submitted)

Problem 1: Write a program to find the range of a list of integers entered through the keyboard. Range is the difference between the biggest and the smallest integers in the list.

Note: Read the size of the list followed by the values (both list size and list values are integers). The list size should be validated such that an error message “Invalid input” to be displayed if the value entered is zero or a negative integer. Both positive and negative integers should be allowed for the list values.

Sample Input 1:

-4

Output 1:

Invalid input

Sample Input 2:

4

-9 1 2 3

Output 2:

12

Sample Input 3:

6

2 2 2 2 2

Output 3:

0

Problem 2: Write a program to receive an integer and find the reverse of the given integer (ignore the leading zeros in the output).

Note: The entered input should be a non-negative integer. In case of a negative integer received from keyboard, an error message “Invalid input” should be displayed.

Sample Input 1:

1260

Output 1:

0621

Sample Input 2:

-121

Output 2:

Invalid input

Problem 3: Write a program to receive an integer and find the octal equivalent of the given integer.

Hint: To obtain octal equivalent of an integer, divide it continuously by 8 till dividend doesn't become zero, then write the remainders obtained in reverse direction.

Note: The entered input should be a non-negative integer. In case of a negative integer received from keyboard, an error message "Invalid input" should be displayed.

Sample Input 1:

9

Output 1:

11

Sample Input 2:

-121

Output 2:

Invalid input

Sample Input 3:

8888

Output 3:

21270

Problem 4: Write a program to receive an integer, n and output the count of number of positive integers less than n that are co-prime to n .

Hint: Two integers are said to be co-prime if their greatest common divisor (GCD) is 1. Co-prime count for 1 is 1.

Note: The entered input should be a positive integer. In case of zero or a negative integer received from keyboard, an error message "Invalid input" should be displayed.

Sample Input 1:

-9

Output 1:

Invalid input

Sample Input 2:

24

Output 2:

8

Sample Input 3:

13

Output 3:

12

Part B (need not be submitted)

1: Here is an example of a (C) program written by a **careless** programmer.

```
void main()
{
    int i, number = -5, value = 10;
    if (number >= 0)
        for (i = 0; i < number; i++)
            if (value == 10)
            {
                printf("Value correct");
                return 0;
            }
    else
        printf("Error: number is negative");
    return 0;
}
```

What is the output after its execution?

(1) Value correct (2) Error: number is negative (3) Compilation Error (4) None of the above

2: The piece of (C) code : **for (;;) { }** is an example of :

(1) Compilation Error (2) A do-nothing operation (3)  Infinite Loop (4) Dummy function

3: What is the output of the following (C) program? _____

```
void main()
{
    int i = 10;
    {
```

```

    int i = 20;
    {
        int i = 30;
    }
    printf("%d", i);
}

```

Handwritten annotations: A blue arrow points from the first `int i = 20;` to the `printf("%d", i);` line, with the number `20` written next to it.

4: What is the output when the following piece of (C) code is executed?

```

int a[5] = {5, 1, 15, 20, 25};
int i, j, k;
i = ++a[1];
j = a[1]++;
k = a[1] * a[2];
printf("%d, %d, %d", i, j, k);

```

Handwritten annotations: A blue checkmark is next to the first line of the code block. The number `2` is written next to `++a[1]`, `2` next to `a[1]++`, and `45` next to `a[1] * a[2]`.

(1) 2, 2, 45 (2) 2, 2, 3 (3) 1, 2, 45 (4) 2, 2, 30

Handwritten annotations: A blue checkmark is next to option (1).

5: What is the output when the following piece of (C) code is executed?

```

int i;
for (;;) {
    printf("TESTING\n");
    break;
}

```

(1) No output (2) A large number of "TESTING"
 (3) TESTING (4) Compilation error

Handwritten annotations: A blue checkmark is next to option (3).

6: Write a program for a matchstick game being played between the computer and a user. Your program should ensure that the computer always wins. Rules for the game are as follows:

- There are 21 matchsticks.
- The computer asks the player to pick 1, 2, 3, or 4 matchsticks.
- After the person picks, the computer does its picking.
- Whoever is forced to pick up the last matchstick loses the game.