

Victoria University of Bangladesh

MD. AJ - Amin

ID: 2121210041

Course: CSI 121 (Structured of Programming
Language)

Dept. of CSE

Final examination - Fall 2021.

Answer to the Question NO: 1 (A)

A function in C programming is a block of code that perform a specific task, can accept input values.

Syntax for defining a function is:

```
return_type function_name (parameter_list)
```

```
{  
    /* function body */  
}
```

Answer to the Question 1 (B)

```

#include <stdio.h>
int find largest (int a, int b, int c) {
    if (a >= b and a >= c) {
        return a;
    } else if (b >= a and b >= c) {
        return b;
    } else {
        return c;
    }
}

int main () {
    int num1, num2, num3;
    printf ("Enter three numbers");
    scanf ("%d %d", &num1, &num2, &num3);
    printf ("The largest number is : %d\n", find
largest (num1, num2, num3));
    return 0;
}

```

Answer to the Question No (1)c.

1. advantage of using function.

- Functions help in breaking down complex problems into simpler pieces.
- They promote code reusability.
- Functions make the code easier to read, maintain and debug.

Answer to the Question NO 2(A)

An array is a collection of elements, of the same type stored in contiguous memory location. Ex -

```
int numbers[5] = {1, 2, 3, 4, 5};
```

Answer to the Question NO 2(B)

```
#include <stdio.h>
int main() {
    int marks[30], i, sum = 0;
    float average;
    printf("Enter marks for 30 students: \n");
    for (i = 0; i < 30; i++) {
        scanf("%d", &marks[i]);
        sum += marks[i];
    }
    average = sum / 30.0;
    printf("The average marks are: %.2f\n", average);
    return 0;
}
```

Answer to the Question No 2 (a).Advantages:

- (i) Arrays allow - random access elements.
- (ii) They are useful for storing multiple values of the same type.

Disadvantages:

- (i) The size of an array cannot be changed once declared.
- (ii) Inefficient insertion and deletion.

Answer to the Question NO: 3(A).

```
#include <stdio.h>
int gcd (int a, int b) {
    if (b == 0)
        return a;
    return gcd (b, a % b);
}
int main () {
    int num1, num2;
    printf ("%d %d", num1, num2);
    printf ("\n GCD of %d and %d is %d\n",
    num1, num2, gcd (num1, num2));
    return 0;
}
```

Answer to the Question NO: 3(B).

```
# include <stdio.h>
```

```
int main () {
```

```
int sum = 0, i = 225;
```

```
do {
```

```
sum += i;
```

```
i += 2;
```

```
} while (i <= 225);
```

```
printf ("The sum of the series is: %d\n",  
sum);
```

```
return 0;
```


Answer to the Question NO-4(A).

Program to Check Leap Year:

```
#include <stdio.h>
```

```
int main() {
```

```
    printf("Enter a year:");
```

```
    scanf("%d", &year);
```

```
    // leap year is perfectly divisible by 400
```

```
    if (year % 400 == 0) {
```

```
        printf("%d is a leap year.", year);
```

```
    }
```

```
    // not a leap year is divisible by 100.
```

```
    // but not divisible by 400.
```

```
    else if (year % 100 == 0) {
```

```
        printf("%d is not a leap year.", year);
```

```
    }
```

```
    // leap year is not divisible by 100
```

```
    // but divisible by 4
```

```
    else if (year % 4 == 0) {
```

```
        printf("%d is a leap year.", year);
```

```
    }
```

```
    // all other years are not leap years
```

```
    else {
```

```
        printf("%d is not a leap year.", year);
```

}

return 0;

}

Output 1

Enter a year: 1900.

It's not a leap year.

Output 2

Enter a year: 2012

2012 is a leap year.

Answer to the Question NO: 4(B)

```
#include <stdio.h>
int gcd (int a, int b) {
    if (b == 0)
        return a
    return gcd (b, a % b);
}
int lcm (int a, int b) {
    return (a * b) / gcd (a, b).
}
int main () {
    int num1, num2;
    printf ("Enter two positive integers: ")
    scanf ("%d %d", &num1, &num2);
    printf ("LCM of %d and %d is %d\n",
    num1, num2, lcm(num1), num2);
    return 0;
}
```