Q1: Give the output of the following snippets, or indicate error with a short explanation. [Marks: 3]

1.

```
int a=1, b=2, c=3, d=4, e=4;
((((a=b)=c)=d)=e)=0;
printf("%d %d %d %d %d", a, b, c, d, e);
```

Output: Compilation error, require some literal on the left of assignment

2.

```
int a=1, b=2, c=3, d=4, e=4, f=6;
f = ((((a=b)==c)==d)==e)==0;
printf("%d %d %d %d %d %d", a, b, c, d, e, f);
```

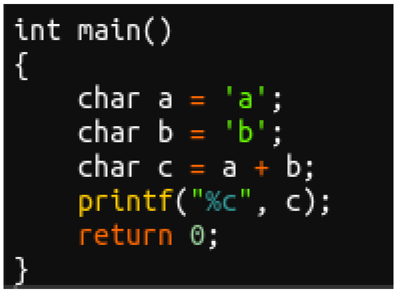
Output: 2 2 3 4 4 1

3.

```
int a=1, b=2, c=3, d=4, e=5, f=6;
f = a = (b = 4*(c = 3*(d = 2*(e = 1))));
printf("%d %d %d %d %d %d", a, b, c, d, e, f);
```

Output: 24 24 6 2 1 24

Q2: What will be the output of the following code? [Marks: 2]



```
int main()
{
    char a = 'a';
    char b = 'b';
    char c = a + b;
    printf("%c", c);
    return 0;
}
```

Options:

A. Compilation Error: Because you can't add two characters.

B. Another valid character.

C. Character not in the English alphabet

D. 'b'

Answer: B and C.

Q3: Determine the size of the array, initialized using the declaration below, following the below rules. [Marks: 2]

1. enter 'ERROR' in case of wrong initialisations
2. include null character in length of char arrays.

A. int arr[5] = {1,2,3,4,5};

B. int arr[10] = {1,2,3};

C. int arr[3] = {1,2,3,5};

D. char arr[] = {'I',' ','a','m',' ','D','r','.',' ','S','t','r','a','n','g','e','\0'};

Ans : 5,10,ERROR,17

Q4: You are given a function func() with prototype int func(int x). Consider the following program, [Marks: 1]

```
int main(){
   int x = 0;
   func(x == 1);
   return 0;
}
```

Will this lead to a compilation error?

Answer: False

Q5: A square matrix `M` with `N` rows is represented in an array A such that `M[i][j]` comes before `M[p][q]` in the array if and only if `i < p` or `i = p` and `j < q`. Then what is the position of `M[y][x]` in the array? [Marks: 2]

Note: M[i][j] denotes the element in the ith row and jth column of the matrix M.

(A) x * N + y

(B) (y-1)* N + x

(C) y * N + y

(D) (x-1) * N + x

Answer: **(B)**

Q6: Not passing arguments to a function defined as int foo(int a) in the caller program will give a compilation error. [Marks: 1]

TRUE

Q7: Passing a long value to a function defined as int foo(int a) in the caller program will not give a compilation error. [Marks 1]

(TRUE

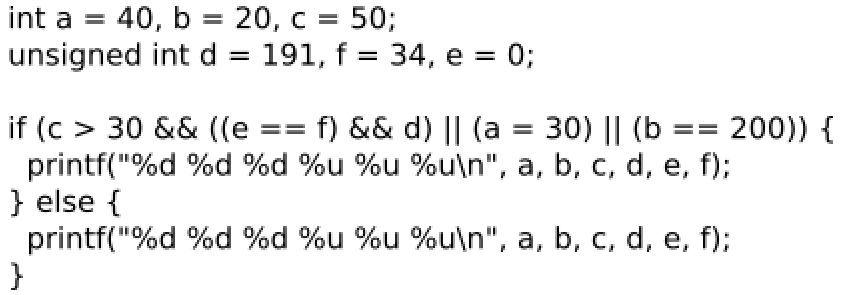
Q8: Select the correct option regarding the output of the following snippet. [Marks: 2]

```
int main()
{
        int i = 1;
        while(i <= 10);
        {
                if(i == 5)
                {
                        break;
                }
                i += 2;
        }
        printf("%d",i);
}
```

OPTION A : Infinite Loop
OPTION B : 7
OPTION C : 5
OPTION D : Compile error

ANSWER: OPTION D

Q9: Output of the given snippet. [Marks: 2]



```
int a = 40, b = 20, c = 50;
unsigned int d = 191, f = 34, e = 0;

if (c > 30 && ((e == f) && d) || (a = 30) || (b == 200)) {
  printf("%d %d %d %u %u %u\n", a, b, c, d, e, f);
} else {
  printf("%d %d %d %u %u %u\n", a, b, c, d, e, f);
}
```

Answer: 30 20 50 191 0 34

Q10: For the given function which takes an array nums and two integers n (length of array) and k. The array is assumed to be sorted in ascending order (nums[i] <= nums[j] for all i < j). [Marks: 3]

```
int foo(int nums[], int n, int k)
{
   int i = 0, j = 0, cnt = 0;
   for(i = 0; i<n-1; i++)
   {
      while(j < n && nums[j]-nums[i] <= k)
         j++;
      cnt += j-i-1;
   }
   return cnt;
}
```

Give function output for following inputs:

1. nums[ ] = {1, 1, 2, 2}, n = 4, k = 1
   - Answer: **6**
2. nums[ ] = {1, 2, 5, 8, 9}, n = 5, k = 2
   - Answer: **2**

Q11: Which of the following are valid variable names in C? (More than one may be correct) [Marks: 1]

E. var_1?

F. var&1

G. _1var

H. __

Answer: GH

Q12: Find the Input? [Marks: 3]

```
#include<stdio.h>
int main(){
    char a[8] = “______”;
    char b[8];
    b[7] = ‘\0’;
    for(int i = 6; i >= 0; i--){
            b[6-i] = a[i] + ‘A’ - ‘a’;
    }
    printf(“%s”, b);
    return 0;
}
```

How should the character array a be initialized so that the output of the above code is CSEIITK?

Answer: ktiiesc

Q13: Consider the code snippet given below. [Marks: 2]

```
int c = <_input_>;
while((c <= 123456) && !('3' == c + 8) && (c += 2));
printf("%d", c);
```

For the following values of <_input_>, what will be the output printed? If the program does not terminate, write INFINITE_LOOP.

(a) -7

Answer: 43

(b) -14

Answer: 0

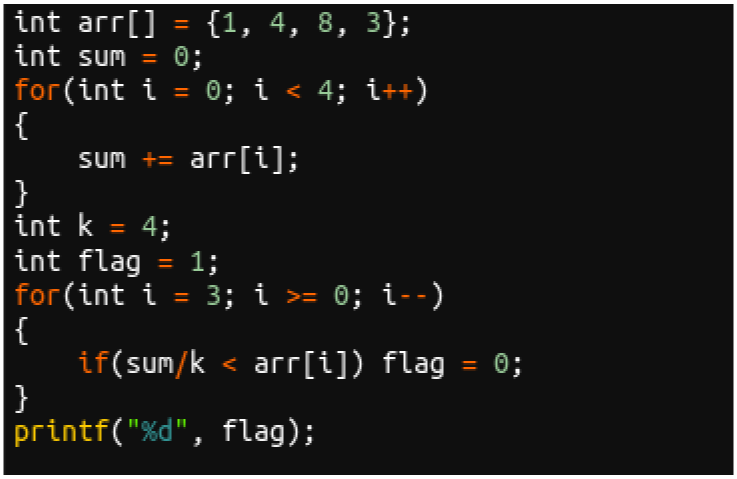
(c) 21

Answer: 43

(d) 42

Answer: 123458

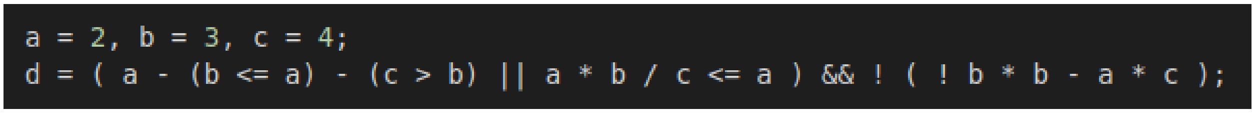
Q14: Predict the Output [3 Marks]



```
int arr[] = {1, 4, 8, 3};
int sum = 0;
for(int i = 0; i < 4; i++)
{
    sum += arr[i];
}
int k = 4;
int flag = 1;
for(int i = 3; i >= 0; i--)
{
    if(sum/k < arr[i]) flag = 0;
}
printf("%d", flag);
```

Answer : 0

Q15: Consider the following where, a,b,c,d are integers. [2 Marks]



```
a = 2, b = 3, c = 4;
d = ( a - (b <= a) - (c > b) || a * b / c <= a ) && ! ( ! b * b - a * c );
```

What is the value of d?

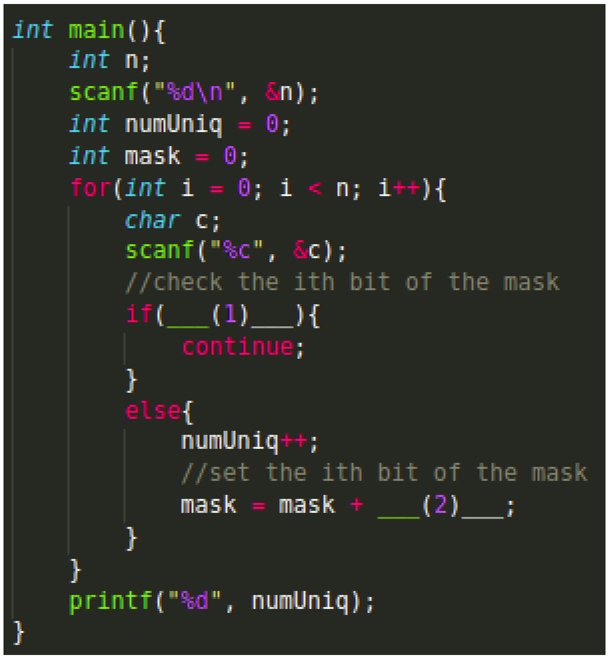
(a) true
(b) false
(c) 0
(d) 1

Answer: (c) 0

Q16: The given code checks the number of unique letters in a string comprising of only LOWERCASE alphabetical characters. [6 Marks]

Hint : You are ALLOWED to use the library function pow(). The logic is to use a mask to check if a letter has appeared yet or not. The ith bit of the mask tells us if the ith lowercase character has appeared or not. If it has appeared, we do nothing, otherwise we increment the count of the unique letters and set the ith bit.

Your task is to fill in the blank with code to achieve the above specification.



```
int main(){
    int n;
    scanf("%d\n", &n);
    int numUniq = 0;
    int mask = 0;
    for(int i = 0; i < n; i++){
        char c;
        scanf("%c", &c);
        //check the ith bit of the mask
        if(___(1)___){
            continue;
        }
        else{
            numUniq++;
            //set the ith bit of the mask
            mask = mask + ___(2)___;
        }
    }
    printf("%d", numUniq);
}
```

Answer:

1. (mask / (int)pow(2, c-'a')) % 2 == 1
2. (int)pow(2, c-'a')

Q17: You are given the following 'incomplete' code: [3X2 = 6 Marks]

```
        #include <stdio.h>
        int main(){
        int n;
        scanf("%d", &n);
        for(int  i=1; i<=2*n-1; i++){
                    for(int  j=1; j<=n; j++){
                            if(i<=n){
                                    if(_____(a)_____ <= n) printf(" ");
                                    else  printf("%d", _____(b)_____);
                            }
                            else{
                                    if (i-j>=n) printf(" ");
                                    else  printf("%d", _____(c)_____);
                            }
                    }
                    printf("\n");
            }
        return 0;
    }
```

Also, you are given the following three expressions: (1) n-i+j (2) i+j (3) i+j-n

Which of the following options maps the expressions to their correct positions in the code such that when given input 4, the code generates the output

```
   1
  12
 123
1234
 123
  12
   1
```

(A) (a)-(2), (b)-(1), (c)-(3)

(B) (a)-(2), (b)-(3), (c)-(1)

(C) (a)-(3), (b)-(1), (c)-(2)

(D) (a)-(3), (b)-(2), (c)-(1)

Answer: **(B)**

Q18: [3x2 = 6 Marks]

```
int trimulti(int n):
        if (n==3) return 3;
        else if(n==2) return 2;
        else if(n==1) return 1;
        else return trimulti(n-1)*trimulti(n-2)*trimulti(n-3);
```

(a) If n=4 is passed to the function, for the 3 different function calls in the function, 3 different respective return value spaces will be allocated in the stack.

Answer: T

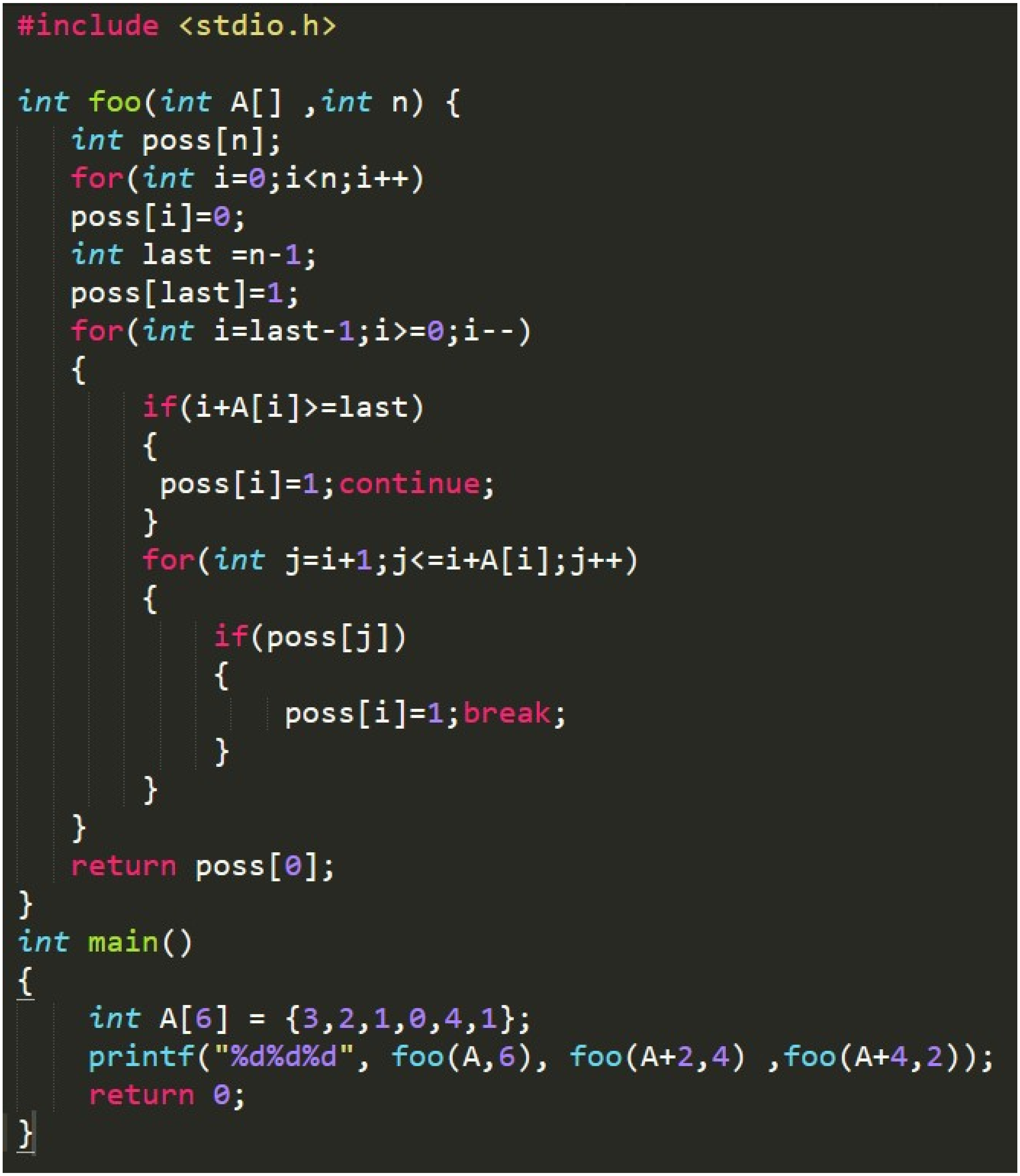
(b) The stack space required to execute trimulti(3) will be more space required for trimulti(2).

Answer: F

(c) How many times will trimulti() be required to be called, if f(4) is needed to be calculated?

Answer: 4

Q19: What will be the output for the following code? [6 Marks]



```c
#include <stdio.h>

int foo(int A[] ,int n) {
    int poss[n];
    for(int i=0;i<n;i++)
    poss[i]=0;
    int last =n-1;
    poss[last]=1;
    for(int i=last-1;i>=0;i--)
    {
        if(i+A[i]>=last)
        {
         poss[i]=1;continue;
        }
        for(int j=i+1;j<=i+A[i];j++)
        {
            if(poss[j])
            {
                poss[i]=1;break;
            }
        }
    }
    return poss[0];
}
int main()
{
     int A[6] = {3,2,1,0,4,1};
     printf("%d%d%d", foo(A,6), foo(A+2,4) ,foo(A+4,2));
     return 0;
}
```

Answer: 001

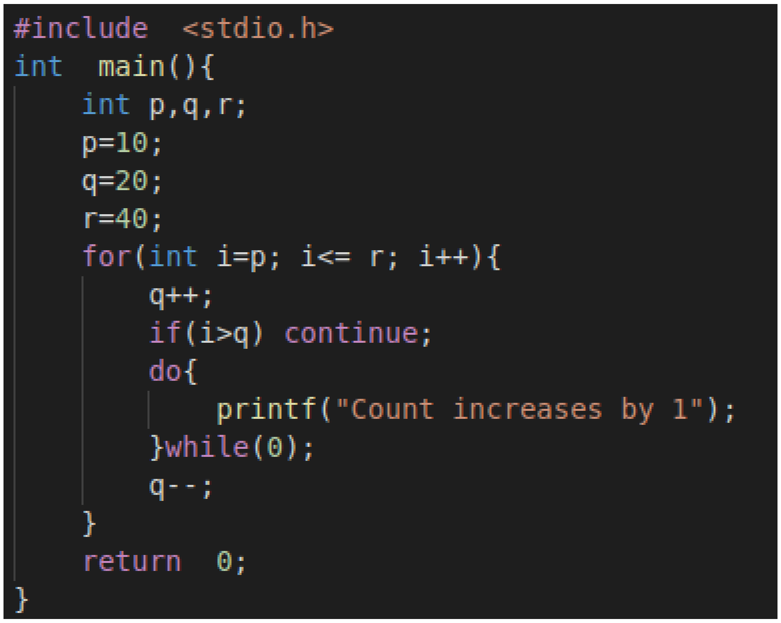
Q20**:** Find the Output [4+2=6 Marks]

```
int a[] = {1,2,3,4,5,6,7,8,9,0};

for(int i = 0; i < 10; i++){

        if(i < 7 && i++%3 == 0)

                a[i++] = 0;

}
```

How many non-zero elements are there in the array a, after executing the snippet? Does the answer change for multiple executions of the for loop on a?

Answer - 6, no

Q21**:** Find how many times the 'do-while' loop gets executed in the following code snippet. [5 Marks]



```
#include  <stdio.h>
int  main(){
    int p,q,r;
    p=10;
    q=20;
    r=40;
    for(int i=p; i<= r; i++){
        q++;
        if(i>q) continue;
        do{
            printf("Count increases by 1");
        }while(0);
        q--;
    }
    return  0;
}
```

Answer: **12**

Q22: Rohit wrote a code to evaluate the value of a mathematical expression. However, the code turned out to be wrong. Given the following inputs, write the output of this code. Also explain where Rohit went wrong in at most two sentences. Note that you do not need to give a modification, just point out the error. [10 Marks]

The variable “eqn” is a string. When we pass strings into a function, we need pointers, which you will learn later in this course. For now, think of “eqn” and “*eqn” as arrays, and the eqn[i] gives us the (i+1)-th character, as is the case with usual arrays you are familiar with.

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

int func1(char c) { return (c >= '0' && c <= '9'); }

int func2(char c) { return (c - '1'); }

int evaluate(char *eqn)
{
    if (*eqn == '\0')  return -1;
    int val = func2(eqn[0]);
    for (int i = 1; eqn[i]; i += 2)
    {
        char opr = eqn[i], opd = eqn[i+1];
        if (!func1(opd))  return -1;
        if (opr == '+')       val += func2(opd);
        else if (opr == '-')  val -= func2(opd);
        else if (opr == '*')  val *= func2(opd);
        else if (opr == '/')  val /= func2(opd);
        else                  return -1;
    }
    return val;
}
int main()
{
    char eqn[500];
    scanf("%s", eqn);
    int val = evaluate(eqn);
    (val == -1)? printf("%s is Invalid", eqn):
                printf("Value of %s is %d", eqn, val);
            return 0;
}
```

Write the outputs according to the given inputs. [6 Marks]

1. Input: 3+7/3+5
   Output: Value of 3+7/3+5 is 8
2. Input: 5+4*2/8
   Output: Value of 5+4*2/8 is 1
3. Input: 9+6/4*3
   Output: Value of 9+6/4*3 is 8
4. Input: 4++1
   Output: 4++1 is Invalid

Write your explanations which resulted in the code being erroneous: [2+2 Marks]

- Line 5: It should be (c - ‘0’) instead of (c-’1’).

- Line 16, 17: Dividing/multiplying by the entire value obtained until now.