**10 MCQ (1 mark each)**

Q.1. Can the fixed arguments passed to the function that accepts variable argument list, occur at the end?

A) Yes

B) No

Q.2. Will the program compile successfully?

#include<stdio.h>

int main()

{

char a[] = "Hello";

char \*p = "Welcome";

a = "Welcome";

p = "Hello";

printf("%s %s\n", a, p);

return 0;

}

A) Yes

B) No

Q.3. What will be the output of the program ?

#include<stdio.h>

#include<string.h>

int main()

{

printf("%d\n", strlen("123456"));

return 0;

}

A) 6

B) 12

C) 7

D) 2

Q.4. In a function that receives variable number of arguments the fixed arguments passed to the function can be at the end of argument list.

A) True

B) False

Q.5. If the two strings are identical, then strcmp() function returns

A) -1

B) 1

C) 0

D) Yes

Q.6. In C, if you pass an array as an argument to a function, what actually gets passed?

A) Value of elements in array

B) First element of the array

C) Base address of the array

D) Address of the last element of array

Q.7. Will the program compile?

#include<stdio.h>

int main()

{

char str[5] = "WelcomeIndia";

return 0;

}

A) Yes

B) No

Q.8. The operator used to get value at address stored in a pointer variable is

A) \*

B) &

C) &&

D) ||

Q.9. What will be the output of given code?

main()

{

char str[7] = "strings";

printf("%s", str);

}

A. error

B. strings

C. cannot predict

D. none of the above

Q.10. The statement that transfers control to the beginning of the loop is called \_\_\_\_\_\_\_\_\_\_.

A. break statement

B. exit statement

C. continue statement

D. goto statement

**5 MCQ (2 mark each)**

Q.1. What will be the output of the program ?

#include<stdio.h>

int main()

{

printf("Hi", "Hello\n");

return 0;

}

A) Error

B) Hi Hello

C) Hi

D) Hello

Q.2. What will be the output of the program If characters 'a', 'b' and 'c' enter are supplied as input?

#include<stdio.h>

int main()

{

void fun();

fun();

printf("\n");

return 0;

}

void fun()

{

char c;

if((c = getchar())!= '\n')

fun();

printf("%c", c);

}

A) abc abc

B) bca

C) Infinite loop

D) cba

Q.3. What will be the output of the program ?

#include<stdio.h>

#include<string.h>

int main()

{

char str[] = "Hello\0\World\0";

printf("%s\n", str);

return 0;

}

A) World

B) Hello

C) Hello World

D) Hello\0World

Q.4. Point out the error in the program

#include<stdio.h>

int f(int a)

{

a > 20? return(10): return(20);

}

int main()

{

int f(int);

int b;

b = f(20);

printf("%d\n", b);

return 0;

}

A) Error: Prototype declaration

B) No error

C) Error: return statement cannot be used with conditional operators

D) None of above

Q.5. Will the program compile in Turbo C?

#include<stdio.h>

int main()

{

int a=10, \*j;

void \*k;

j=k=&a;

j++;

k++;

printf("%u %u\n", j, k);

return 0;

}

A) Yes

B) No

**2 Coding Questions (5 mark each)**

Q.1. Number Recognition

Company called 'MathGenius' has designed a course for called Learning Number Recognition and Counting. The assessment part of the course has a question where the student is given a number and a digit. The student needs to find out the total count of the digits present in the number excluding the given digit.

Write a c program to help the student find out the count of the total number of digits present in the number excluding the given digit.

**Sample Input 1**

5644456 5

**Sample Output 1**

5

**Sample Input 2**

55555 5

**Sample Output 2**

0

**Input Explanation**

The input consists of two space-separated integers – number and digit,

where the first integer represents the number and the second integer represents the digit given to the student.

**Output Explanation**

Print an integer representing the count of the total number of digits present in the number excluding the given digit.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 775642156 6 | 23145645 7 | 443475 4 | 5644456 5 | 111111111 1 |
| **Output** | 7 | 8 | 3 | 5 | 0 |

**#Solution**

#include<stdio.h>

int excludingDigit(int,int);

int main()

{

int num,n;

scanf("%d %d",&num,&n);

printf("%d",excludingDigit(num,n));

return 0;

}

int excludingDigit(int num,int n)

{

int count=0,digit;

while(num)

{

digit=num%10;

num=num/10;

if(digit!=n)

{

count++;

}

}

return count;

}

Q.2. Perfect cube

The children's toy-making company "ToysFun" is building cubic-shaped learning toys. The company has a list of N dimensions suggested by its designers but they wish to choose only those dimensions for the toys that are perfect cube numbers. To do this, they need to know the total count of perfect cube numbers present in the list of dimensions.

Write a function to help the toy manufacturers find the total count of perfect cube numbers present in the list of dimensions.

Example

Input

9

23 1 8 56 27 67 64 125 232

Output

5

Explanation

The cube numbers are 1, 8, 27, 64, 125. Hence the output is 5.

**Sample Input 1**

5

23 1 8

**Sample Output 1**

3

**Sample Input 2**

3

54 96 81

**Sample Output 2**

0

**Input Explanation**

The first line of input consists of an single integer, representing the total number of dimensions selected by the designers (N).

The second line of input consists of N space-separated integers, representing the value of the dimensions selected by the designers.

**Output Explanation**

Print an integer value representing the total count of the dimensions that are perfect cube numbers.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 3  27 67 64 | 4  8 9 27 2 | 5  23 1 8 | 3  4 9 12 15 | 5  1 27 8 125 512 |
| **Output** | 2 | 3 | 3 | 0 | 5 |

**#Solution**

#include<stdio.h>

int dimens\_is\_cube(int n)

{

int i=1;

int result = 0;

while(i<=n)

{

if(i\*i\*i == n)

{

result = 1;

break;

}

i++;

}

return result;

}

int main()

{

int numDimensions;

scanf("%d", &numDimensions);

int dimens[numDimensions];

for(int i=0;i<numDimensions;i++)

{

scanf("%d", &dimens[i]);

}

int perfect\_cube = 0;

for(int i=0;i<numDimensions;i++)

{

if(dimens\_is\_cube(dimens[i]))

{

perfect\_cube++;

}

}

printf("%d",perfect\_cube);

return 0;

}

**1 Coding Question (10 mark)**

Q.1. Shooting Game

The games development company "FunGames" has developed a balloon shooter game. The balloons are arranged in a linear sequence and each balloon has a number associated with it. The numbers on the balloons are in the Fibonacci series. In the game, the player shoots 'k' balloons. The player's score is the sum of numbers on the 'k' balloons. Write a program to generate the player's score.

Example

Input

7

Output

20

Explanation

The Fibonacci sum is 0+1+1+2+3+5+8=20

**Sample Input 1**

8

**Sample Output 1**

33

**Sample Input 2**

0

**Sample Output 2**

0

**Input Explanation**

The input consists of an single integer, representing the total number of balloons shot by the player (k).

**Output Explanation**

Print an integer value representing the player's score. If no balloons are shot then print 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 9 | 8 | 4 | 0 | 10 |
| **Output** | 54 | 33 | 4 | 0 | 88 |

**#Solution**

#include<stdio.h>

int fibo(int n)

{

int fibs[n];

fibs[0] = 0;

fibs[1] = 1;

int sum=0;

for(int i=2;i<n;i++)

{

fibs[i] = fibs[i-1]+fibs[i-2];

}

for(int i=0;i<n;i++)

{

sum += fibs[i];

}

return sum;

}

int main()

{

int numBalloons;

scanf("%d",&numBalloons);

if (numBalloons==0)

{

printf("0");

}

else

{

int result = fibo(numBalloons);

printf("%d",result);

}

}