**10 MCQ (1 mark each)**

Q.1. The code printf("%d",printf("tim")); \_\_\_\_\_\_\_\_\_\_.

A) results in a syntax error

B) output is tim3

C) outputs a garbage value

D) prints tim and terminates abruptly

Q.2. The prototypes of all standard library string functions are declared in the file string.h.

A) Yes

B) No

Q.3. Is there any difference between the two statements?

char \*ch = "WELCOME";

char ch[] = "welcome";

A) Yes

B) No

Q.4. What will be the output of the program ?

#include<stdio.h>

#include<string.h>

int main()

{

char str[] = "India\0\BIX\0";

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

return 0;

}

A) 10

B) 6

C) 5

D) 11

Q.5. What will be the output of the program ?

#include<stdio.h>

int main()

{

char str[7] = "HelloWorld";

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

return 0;

}

A) Error

B) HelloWorld

C) Cannot predict

D) None of above

Q.6. Can we pass a variable argument list to a function at run-time?

A) Yes

B) No

Q.7. What will sizeof(myArray) in the following type definition? (Assume one character occupies 1 byte)

typedef char x[10];

x myArray[5];

A) 15 bytes

B) 10 bytes

C) 50 bytes

D) 30 bytes

Q.8. If the two strings are found to be unequal then strcmp returns difference between the first non-matching pair of characters.

A) True

B) False

Q.9. The following program \_\_\_\_\_\_\_\_\_\_.

main()

{

static int a[] = {7,8,9};

printf("%d", 2[a] + a[2]);

}

A) results in bus error

B) results in segmentation violation error

C) will not compile successfully

D) none of the above

Q.10. Can you use the fprintf() to display the output on the screen?

A) Yes

B) No

**5 MCQ (2 mark each)**

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

#include<stdio.h>

int main()

{

char \*names[] = { "Suresh", "Siva", "Sona", "Baiju", "Ritu"};

int i;

char \*t;

t = names[3];

names[3] = names[4];

names[4] = t;

for(i=0; i<=4; i++)

printf("%s,", names[i]);

return 0;

}

A) Suresh, Siva, Sona, Baiju, Ritu

B) Suresh, Siva, Sona, Ritu, Baiju

C) Suresh, Siva, Baiju, Sona, Ritu

D) Suresh, Siva, Ritu, Sona, Baiju

Q.2. What will be the output of following code?

#include <stdio.h>

main()

{

register i = 5;

char j[] = "hello";

printf("%s %d", j, i);

}

A) error

B) hello 5

C) hello

D) none of the above

Q.3. The following program fragment results in \_\_\_\_\_\_\_\_\_\_?

for(i = 1; i < 5 ; ++i)

{

if(i == 3)

{

continue;

}

else

{

printf("%d", i);

}

}

A) 1 2 4 5

B) 1 2 4

C) 2 4 5

D) none of the above

Q.4. What will be the output of the following loop?

for(i=1, j=10; i<6; ++i, --j)

{

printf("%d %d", i, j);

}

A) 1 1 0 2 9 3 8 4 7 5 6

B) 1 2 3 4 5 10 9 8 7 6

C) 1 1 1 1 1 9 9 9 9 9

D) none of above

Q.5. In the following code fragment, on termination j will have the value?

i = 6720; j = 4;

while((i%j) == 0)

{

i = i/j;

j = j+1;

}

A) 4

B) 8

C) 9

D) 6720

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

Q.1. Weather forecasting services

"Climate Today" is a media company that provides commercial weather forecasting services. They have a list of temperatures for a particular location for N days and have already prepared the

forecast report within the range L to R where L is the minimum temperature and R is the maximum temperature from the given list. Now, "Climate Today" needs to know the temperature for the

days that are not included within the range L to R.

Write an algorithm to help the company prepare the list of temperatures that are not included in the range L to R.

**Sample Input 1**

7 3 6

2 5 1 8 6 9 4

**Sample Output 1**

2 1 8 9

**Sample Input 2**

6 2 5

5 1 8 6 9 4

**Sample Output 2**

1 8 6 9

**Input Explanation**

The first line of input consists of three integers, representing the number of days(N), the minimum temperature (L), and the maximum temperature (R).

The second line consists of N space-separated integers - temp1, temp2, ... tempN-1 representing the temperature of N days.

**Output Explanation**

Print space-separated integers representing the list of temperatures that are not covered in the range L to R.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 6 3 6  2 5 1 8 9 4 | 6 2 5  6 9 4 2 1 8 | 8 3 6  2 5 1 6 9 4 8 3 | 4 3 1  9 5 2 3 | 6 2 5  5 1 8 6 9 4 |
| **Output** | 2 1 8 9 | 6 9 1 8 | 2 1 9 8 | 9 5 2 3 | 1 8 6 9 |

**#Solution**

#include<stdio.h>

void method(int temp[],int days,int min,int max)

{

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

{

if(temp[i]<min || temp[i]>max)

{

printf("%d ", temp[i]);

}

}

}

int main()

{

int days;

scanf("%d", &days);

int temp[days];

int min,max;

scanf("%d %d", &min, &max);

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

{

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

}

method(temp,days,min,max);

}

Q.2. Count odd and even

Given an array of integers, count the total number of odd elements and even elements in the array and display them as output.

**Sample Input 1**

3

1 2 3

**Sample Output 1**

Odd: 2

Even: 1

**Sample Input 2**

5

1 2 3 4 5

**Sample Output 2**

Odd: 3

Even: 2

**Input Explanation**

Input consists of two lines

First line input represents size of array

Second line will be multiple space separate integer value, that are elements of array

**Output Explanation**

Output consists combination of character and integers in two lines

First line will display count of even

Second line will display count of odd numbers

**Note: make sure there are no extra spaces in output line**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | 4  12 15 16 1 | 6  99 65 1 3 7 33 | 3  33 22 11 | 4  33 74 89 66 | 5  11 2 4 15 8 |
| **Output** | Odd: 2  Even: 2 | Odd: 6  Even: 0 | Odd: 2  Even: 1 | Odd: 2  Even: 2 | Odd: 2  Even: 3 |

**#Solution**

#include<stdio.h>

int main()

{

int n;

scanf("%d", &n);

int arr[n];

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

{

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

}

int count\_odd = 0, count\_even = 0;

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

{

if (arr[i] % 2 == 1)

count\_odd++;

else

count\_even++;

}

printf("Odd: %d", count\_odd);

printf("\nEven: %d", count\_even);

return 0;

}

**1 Coding Question (10 mark)**

Q.1. Anagram or not

Two strings are given as input and those strings have to be checked if they are anagrams or not. Anagram means that both strings contain the same character set, only their order will be different. Therefore, in both strings, the frequency of each letter must be the same.

**Sample Input 1**

act

cat

**Sample Output 1**

Anangram

**Sample Input 2**

team

mat

**Sample Output 2**

Not Anagrams

**Input Explanation**

Input consists of two line separated string value (in lowercase only)

**Output Explanation**

Output consists of single string value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** | **Test Case 4** | **Test Case 5** |
| **Input** | cat  bat | rat  art | den  end | dead  bread | tab  bat |
| **Output** | Not Anagrams | Anagram | Anagrams | Not Anagrams | Anagrams |

**#Solution**

#include <stdio.h>

int check\_anagram(char[], char[]);

int main()

{

char a[100], b[100];

scanf("%s", a);

scanf("%s", b);

if (check\_anagram(a, b) == 1)

printf("The strings are anagrams\n");

else

printf("The strings are not anagrams\n");

return 0;

}

int check\_anagram(char a[], char b[])

{

int first[26] = {0}, second[26] = {0}, c = 0;

while (a[c] != '\0')

{

first[a[c] - 'a']++;

c++;

}

c = 0;

while (b[c] != '\0')

{

second[b[c] - 'a']++;

c++;

}

for (c = 0; c < 26; c++)

{

if (first[c] != second[c])

return 0;

}

return 1;

}