**Roll No…………….. Total No. of Pages:……**

**FUNDAMENTALS OF C PROGRAMMING**

**Time allowed: 90 Minutes Max. Marks: 40**

**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

Q1 What is the difference between a single dimensional array and a multi-dimensional array in C language?

A) A single dimensional array has only one index, a multi-dimensional array has multiple indices \*(Correct option)

B) A multi-dimensional array has only one index, a single dimensional array has multiple indices

C) Both single dimensional arrays and multi-dimensional arrays have only one index

D) None of the above

Q2 What is the purpose of the subscript operator ( [ ] ) in an array in C language?

A) To access elements of an array \*(Correct option)

B) To store elements in an array

C) To compare elements of an array

D) None of the above

Q3 What is the purpose of the pointer arithmetic in C language?

A) To add or subtract a value from a pointer \*(Correct option)

B) To compare two pointers

C) To calculate the size of a pointer

D) None of the above

Q4 What is the difference between a string and an array in C language?

A) A string is a sequence of characters, an array is a collection of variables of the same data type \*(Correct option)

B) An array is a sequence of characters, a string is a collection of variables of the same data type

C) Both strings and arrays are sequences of characters

D) None of the above

Q5 What is the purpose of the sizeof operator in C language?

A) To change the size of an array

B) To find the size of a variable or datatype \*(Correct option)

C) To find the size of a saved program

D) None of the above

Q6 Can you store elements of different data types in a two-dimensional array in C language?

A) Yes

B) No \*(Correct option)

C) Only Character array can store ‘Double’ type values

D) Only Integer array can store a string

Q7 What is the purpose of the sort function in C language when working with arrays?

A) To sort the elements of an array in ascending or descending order \*(Correct option)

B) To access elements of an array

C) To compare elements of an array

D) None of the above

Q8 What is a pointer in C language?

A) A variable that holds the address of another variable \*(Correct option)

B) A variable that holds the value of another variable

C) A variable that holds the name of another variable

D) None of the above

Q9 What is the purpose of the & operator in C language?

A) To get the address of a variable \*(Correct option)

B) To get the value of a variable

C) To compare two variables

D) None of the above

Q10 What is the difference between a pre-decrement and a post-decrement operator in C language?

A) The pre-decrement operator decrements the value of a variable before using it, the post-decrement operator decrements the value of a variable after using it \*(Correct option)

B) The post-decrement operator decrements the value of a variable before using it, the pre-decrement operator decrements the value of a variable after using it

C) Both pre-decrement and post-decrement operators decrement the value of a variable before using it

D) None of the above

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

11. Consider the following code in C:

int i = 1;

while (i <= 5)

{ if (i % 2 == 0)

{ break;

}

printf("%d ", i);

i = i + 2; }

What is the output of the code?

a) 1 3 5 \*(Correct option)

b) 2

c) 1 3

d) 1

12. Can you have multiple functions with the same name in C programming?

a) Yes, if the functions have different parameters \*(Correct option)

b) No, functions must have unique names

c) Yes, if the functions are declared in different files

d) Yes, if the functions are defined using the inline keyword.

13.What is the difference between a pointer and a normal variable in C?

a) A normal variable stores the value of a data type, while a pointer stores the memory address of a variable \*(Correct option)

b) Pointers are always faster than normal variables

c) Pointers are always smaller in size than normal variables

d) Normal variables can only be used inside a function, while pointers can be used globally

14.What will be the output of the following code?

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

printf("%d", \*(\*arr + 1) + \*(\*(arr+1)+1));

a) 3

b) 4

c) 2 \*(Correct option)

d) 6

15.What will be the output of the following code?

char name[10] = "Hello";

char \*ptr = name;

printf("%c", \*(ptr+2));

a) 'H'

b) 'e'

c) 'l' \*(Correct option)

d) "llo"

**SECTION-C(Coding Question) (2x5 marks=5 marks)**

Q16 Chaitanya has a matrix containing all 1 except one position(i,j) where A[i][j] =0. He wants to set all the elements of jth column to 0. He is not able to do it. Help him to perform this task.

**Input:**

The first line contains two integers N and M. N and M are the size of matrices Chaitanya has.

Second line contain a matrix of size M\*N.

**Constraints:**

1 <= N, M <= 1000

0 <= A[i][j] <= 1

It is guaranteed that there will be only one cell having 0.

**Output:**

Print a matrix that satisfies the given conditions.

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 2 3  1 0 1  1 1 1 | 1 0 1  1 0 1 |
| STC2 | 3 2  1 1  1 1  1 0 | 1 0  1 0  1 0 |

**Solution 16:**

#include<stdio.h>

void solve(int n, int m, int a[][m])

{

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

{

for(int j=0;j<m;j++)

{

if(a[i][j] == 0)

{

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

{

a[k][j] = 0;

}

return;

}

}

}

}

int main()

{

int n,m;

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

int a[n][m];

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

{

for(int j=0;j<m;j++)

{

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

}

}

solve(n,m,a);

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

{

for(int j=0;j<m;j++)

{

printf("%d ",a[i][j]);

}

printf("\n");

}

return 0;

}

Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | 4 5  1 1 1 1 1  0 1 1 1 1  1 1 1 1 1  1 1 1 1 1 | 5 5  1 1 1 1 1  1 1 0 1 1  1 1 1 1 1  1 1 1 1 1 | 1 1  0 |
| Output | 0 1 1 1 1  0 1 1 1 1  0 1 1 1 1  0 1 1 1 1 | 1 1 0 1 1  1 1 0 1 1  1 1 0 1 1  1 1 0 1 1  1 1 0 1 1 | 0 |

Q17. Suppose you have given an array of strings. Your job is to list these strings in lexicographical order.

**Input:**

array of strings

**Output:**

print lexicographical order of these strings

Example -

Input-

Enter 3 words: Hi Chitkara

We are CSE

Our exam going on

Output -

In the lexicographical order:

Hi Chitkara

Our exam going on

We are CSE

**Solution**

#include <stdio.h>

#include <string.h>

int main() {

char str[3][50], temp[50];

printf("Enter 3 words: ");

// Getting strings input

for (int i = 0; i < 3; ++i) {

fgets(str[i], sizeof(str[i]), stdin);

}

// storing strings in the lexicographical order

for (int i = 0; i < 3; ++i) {

for (int j = i + 1; j < 3; ++j) {

// swapping strings if they are not in the lexicographical order

if (strcmp(str[i], str[j]) > 0) {

strcpy(temp, str[i]);

strcpy(str[i], str[j]);

strcpy(str[j], temp);

}

}

}

printf("\nIn the lexicographical order: \n");

for (int i = 0; i < 3; ++i) {

fputs(str[i], stdout);

}

return 0;

}

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | Enter 3 words: | Enter 3 words:  0  1  2 | Enter 3 words:  a  aaa  aa |
| Output | In the lexicographical order:  Hi Chitkara  Our exam going on  We are CSE | In the lexicographical order:  0  1  2 | In the lexicographical order:  a  aa  aaa |

**SECTION-D (Coding Question)(1x10 mark=10 mark)**

Q18 **Problem Statement: The IT giant "SoftCompInfo" has decided to transfer its message through the network using a new encryption technique. The company has decided to encrypt the data using the non-prime number concept. The message is in the form of a number and the sum of non-prime digits present in the message is used as the encryption key. Write a function to determine the encryption key.**

Input

45673

Output

10

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 6461 | 1001 | 33512 |
| **Output** | 16 | 0 | 0 |

**#Solution**

#include <stdio.h>

int nonprime(int);

int nonprime(int n)

{

int digit,i,sum=0,m;

while(n!=0)

{

digit=n%10;

n=n/10;

for(i=2;i<digit;i++)

{

if(digit%i==0)

{

sum=sum+digit;

break;

}

}

}

return sum;

}

int main()

{

int n;

scanf("%d",&n);

printf("%d",nonprime(n));

return 0;

}