**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 purpose of a break statement in C language?

A) To skip an iteration in a loop

B) To exit a loop \*(Correct option)

C) To exit a function

D) All of the above

Q2 What is the purpose of the strcpy() function in C language?

A) To copy a string to another string \*(Correct option)

B) To compare two strings

C) To reverse a string

D) None of the above

Q3 What is the purpose of the scanf() function in C language?

A) To read input from the user \*(Correct option)

B) To write output to the screen

C) To manipulate strings

D) None of the above

Q4 What is the purpose of a default case in a switch statement in C language?

A) To handle cases that do not match any other case \*(Correct option)

B) To end a switch statement

C) To perform specific tasks

D) None of the above

Q5 What is the purpose of the strcmp() function in C language?

A) To compare two strings \*(Correct option)

B) To copy a string to another string

C) To reverse a string

D) None of the above

Q6 What is the purpose of the sizeof operator in C language?

A) To determine the size of a data type or variable \*(Correct option)

B) To determine the size of a string

C) To determine the size of an array

D) All of the above

Q7 What is the purpose of the for loop in C language?

A) To repeat a set of statements for a specified number of times \*(Correct option)

B) To repeat a set of statements until a condition is met

C) To execute a set of statements only once

D) None of the above

Q8 What is the syntax for declaring an enumerated data type in C language?

A) enum enumeration\_name;

B) enum {enumeration\_list};

C) enum enumeration\_name {enumeration\_list}; \*(Correct option)

D) None of the above

Q9 What is the purpose of a goto statement in C language?

A) To jump to a specific label in the program \*(Correct option)

B) To exit a loop

C) To exit a function

D) All of the above

Q10 What is the difference between a #define and a constant in C language?

A) #define is a preprocessor directive, constants are variables \*(Correct option)

B) Constants are preprocessor directives, #define is a variable

C) Both #define and constants are preprocessor directives

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)

{ printf("%d ", i);

i = i + 2;

}

What is the output of the code?

a) 1 3 5 \*(Correct option)

b) 2 4 6

c) 1 2 3 4 5

d) 5 4 3 2 1

12.What is the purpose of the return statement in a function in C?

a) To return a value from the function to the calling code \*(Correct option)

b) To receive input data and pass it to the function

c) To specify the number of iterations for a loop

d) To perform mathematical operations

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

int x = 20;

int \*ptr1 = &x;

int \*ptr2 = ptr1;

ptr2 = &x;

\*ptr1 = 30;

printf("%d %d", \*ptr1, \*ptr2);

a) 30 30 \*(Correct option)

b) 30 20

c) 20 30

d) 20 20

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

int arr[2][3] = {{1, 2, 3}, {4, 5, 6}};

printf("%d", arr[0][2]);

a) 1

b) 2

c) 3 \*(Correct option)

d) 4

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

char name[10] = "Hello";

name[0] = 'M';

printf("%s", name);

a) 'H'

b) 'Mello' \*(Correct option)

c) 'o'

d) "Hello"

**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 ith row 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 | 0 0 0  1 1 1 |
| STC2 | 3 2  1 1  1 1  1 0 | 1 1  1 1  0 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<m;k++)

{

a[i][k] = 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 | 1 1 1 1 1  0 0 0 0 0  1 1 1 1 1  1 1 1 1 1 | 1 1 1 1 1  0 0 0 0 0  1 1 1 1 1  1 1 1 1 1  1 1 1 1 1 | 0 |

Q17. Suppose you are given a number in decimal form, your job is to convert the decimal number to binary form.

**Input:**

Input n, is the decimal number. Example 14

n <= 15

**Output:**

Given n, output is a binary number. Example 1110

**Solution:**

#include<stdio.h>

int main()

{

unsigned int i, b[32] = {0};

int j = 31, count = 0;

printf("\nEnter the unsigned integer:");

scanf("%d", &i);

while(i != 0)

{

b[j] = i & 0x00000001;

if(b[j] == 1)

count++;

j--;

i = i >> 1;

}

printf("\nThe number of bits set is %d", count);

printf("\nThe binary representation is ");

for(i = j + 1; j < 32; j++)

printf("%d", b[j]);

return 0;

}

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | 15 | 0 | 12 |
| Output | 1111 | 0000 | 1100 |

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

Q18 **Write a function in C to sort an array of strings using a doubly pointer.**

**Sample Input**:

["apple", "banana", "cherry"] // array element

**Sample Output**:

The sorted array of strings is: apple banana cherry

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | ["apple", "banana", "cherry"] | ["dog", "cat", "elephant"] | ["car", "bus", "train"] |
| Output | The sorted array of strings is: apple banana cherry | The sorted array of strings is: cat dog elephant | The sorted array of strings is: bus car train |

**Solution:**

#include <stdio.h>

#include <string.h>

void sortStrings(char\*\* arr, int n) {

char\* temp;

for (int i = 0; i < n - 1; i++) {

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

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

temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

int main() {

int n;

printf("Enter the number of strings: ");

scanf("%d", &n);

char \*arr[n];

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

arr[i] = malloc(100 \* sizeof(char));

printf("Enter string %d: ", i + 1);

scanf("%s", arr[i]);

}

sortStrings(arr, n);

printf("The sorted array of strings is: \n");

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

printf("%s\n", arr[i]);

}

return 0;

}