**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 Can a pointer hold the address of any data type in C language?

A) Yes \*(Correct option)

B) No

C) Sometimes

D) None of the above

Q2 What is the purpose of the null pointer in C language?

A) To indicate an invalid memory address \*(Correct option)

B) To indicate a valid memory address

C) To compare two pointers

D) None of the above

Q3 Can you change the value stored at an address using a pointer in C language?

A) Yes \*(Correct option)

B) No

C) Depends on Datatypes

D) None of the Above

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

A) An array holds multiple values of the same data type, a pointer holds the address of one value of any data type \*(Correct option)

B) A pointer holds multiple values of the same data type, an array holds the address of one value of any data type

C) Both pointers and arrays hold multiple values of the same data type

D) None of the above

Q5 What is the purpose of the continue statement in C language?

A) To continue a loop \*(Correct option)

B) To exit a loop

C) To transfer control to another part of the program

D) None of the above

Q6 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

Q7 What is the purpose of the switch statement in C language?

A) To execute a set of statements based on the value of a single expression \*(Correct option)

B) To execute a set of statements based on the value of multiple expressions

C) To repeat a set of statements

D) None of the above

Q8 Can you use a pointer to pass variables by reference in C language?

A) Yes \*(Correct option)

B) No

C) Depends on variable’s datatype

D)None of the Above

Q9 What is the difference between a normal variable and a pointer variable in C language?

A) A normal variable holds a value, a pointer variable holds the address of a value \*(Correct option)

B) A pointer variable holds a value, a normal variable holds the address of a value

C) Both normal variables and pointer variables hold a value

D) None of the above

Q10 Can you declare an array of pointers in C language?

A) Yes \*(Correct option)

B) No

C) Only integer pointer is allowed

D) None of the above

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

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

11. What is the difference between a while loop and a do-while loop in C?

a) Both while and do-while loops are the same

b) The do-while loop runs at least once, while the while loop runs only if the condition is true \*(Correct option)

c) The while loop runs at least once, while the do-while loop runs only if the condition is true

d) The do-while loop can only be used inside a while loop

12. What is the difference between a library function and a user-defined function in C?

a) Library functions are provided by the C library, while user-defined functions are created by the user \*(Correct option)

b) Library functions are slower than user-defined functions

c) Library functions can only be used inside a user-defined function

d) Library functions can only be used once in a program

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

int x = 10;

int \*ptr1 = &x;

int \*ptr2 = ptr1;

int y = \*ptr2 + \*ptr1;

printf("%d", y);

a) 20 \*(Correct option)

b) 10

c) 30

d) 40

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

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

printf("%d", \*(\*(arr+2)+0));

a) 1

b) 2

c) 3

d) 7 \*(Correct option)

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

char name[10] = "Hello";

printf("%d", sizeof(name) + srlen(name));

a) 10

b) 15 \*(Correct option)

c) 6

d) 4

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

Q16 Chaitanya has an array A. He wants to find how many integers x, present in the array such that the number of integers greater than x in the array equals to x.

**Input:**

The first line contains an integer N giving the size of the array.

Second line has an integer array A of size N

**Constraints:**

1<=**n**<=10^5

0<= **A[i]** <=50

**Output:**

Print number of x in the array.

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | 5  1 2 3 4 5 | 0 |
| STC2 | 4  2 1 3 3 | 1 |

**Solution 16:**

#include<stdio.h>

int solve(int n, int a[],int b[])

{

for(int i=0;i<51;i++)

{

int count=0;

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

{

if(a[j] > i)

{

count++;

}

}

b[i] = count;

}

int ans=0;

for(int i=0;i<51;i++)

{

if(b[i] ==i)

{

ans++;

}

}

return ans;

}

int main()

{

int b[51];

for(int i=0;i<51;i++)

{

b[i] = -1;

}

int n,j;

scanf("%d",&n);

int a[n];

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

{

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

}

int ans = solve(n,a,b);

printf("%d",ans);

return 0;

}

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | 5  13 22 23 49 50 | 3  1 1 2 | 10  0 1 1 1 1 1 1 1 1 6 |
| Output | 0 | 2 | 8 |

Q17. The transpose of a matrix is a new matrix that is obtained by exchanging the rows and columns.

Write a program, where the user is asked to enter the number of rows r and columns c. Their values should be less than 10 in this program. Then, the user is asked to enter the elements of the matrix (of order r\*c).

Given the above entered matix, display the transpose of it.

**Input:**

r = number of rows

c = number of columns

r\*c = total numbers of element in matrix

**Constraints:**

1 <= r <= 10

1 <= c <= 10

**Output:**

transpose of the matrix

Example -

Entered matrix : 1 2 3

Transpose of the matrix:

1

2

3

**Solution**

#include <stdio.h>

int main() {

int a[10][10], transpose[10][10], r, c;

printf("Enter rows and columns: ");

scanf("%d %d", &r, &c);

// assigning elements to the matrix

printf("\nEnter matrix elements:\n");

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

for (int j = 0; j < c; ++j) {

printf("Enter element a%d%d: ", i + 1, j + 1);

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

}

// printing the matrix a[][]

printf("\nEntered matrix: \n");

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

for (int j = 0; j < c; ++j) {

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

if (j == c - 1)

printf("\n");

}

// computing the transpose

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

for (int j = 0; j < c; ++j) {

transpose[j][i] = a[i][j];

}

// printing the transpose

printf("\nTranspose of the matrix:\n");

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

for (int j = 0; j < r; ++j) {

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

if (j == r - 1)

printf("\n");

}

return 0;

}

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | Entered matrix:  1 4 0  -5 2 7 | Entered matrix: 1 | Entered matrix: 1 2 3 |
| Output | Transpose of the matrix:  1 -5  4 2  0 7 | Transpose of the matrix:  1 | Transpose of the matrix:  1  2  3 |

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

Q18 **Problem Statement: Write a function in C that takes in a single dimensional array of integers and its size as input, and returns the sum of all the even numbers in the array. The function should use pointers to traverse the array and access its elements.**

**Sample Input**:

arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

size = 10

**Sample Output**:

The sum of even numbers is 30

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  size = 10 | arr = [11, 12, 13, 14, 15, 16, 17, 18, 19, 20]  size = 10 | arr = [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]  size = 10 |
| Output | The sum of even numbers is 30 | The sum of even numbers is 110 | The sum of even numbers is 0 |

**Solution:**

#include <stdio.h>

int sumOfEvenNumbers(int\* arr, int size) {

int sum = 0;

int\* ptr = arr;

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

if (\*ptr % 2 == 0) {

sum += \*ptr;

}

ptr++;

}

return sum;

}

int main() {

int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int size = sizeof(arr) / sizeof(arr[0]);

int sum = sumOfEvenNumbers(arr, size);

printf("The sum of even numbers is %d\n", sum);

return 0;

}