**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 row-major order and a column-major order in C language when working with multi-dimensional arrays?

A) A row-major order stores elements in rows first, a column-major order stores elements in columns first \*(Correct option)

B) A column-major order stores elements in rows first, a row-major order stores elements in columns first

C) Both have No difference

D) None of the above

Q2 What is the purpose of the strcpy function in C language when working with arrays and strings?

A) To copy elements of one string to another string \*(Correct option)

B) To access elements of a string

C) To compare elements of a string

D) None of the above

Q3 What is the purpose of the strlen function in C language when working with strings and arrays?

A) To find the length of a string \*(Correct option)

B) To access elements of a string

C) To compare elements of a string

D) None of the above

Q4 What is the purpose of the array index in C language when working with arrays?

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

Q5 What is the difference between a static array and a dynamic array in C language?

A) A static array has a fixed size, a dynamic array can change in size \*(Correct option)

B) A dynamic array has a fixed size, a static array can change in size

C) Both static arrays and dynamic arrays have a fixed size

D) None of the above

Q6 What is the difference between a sorted array and an unsorted array in C language?

A) A sorted array has elements in ascending or descending order, an unsorted array does not \*(Correct option)

B) An unsorted array has elements in ascending or descending order, a sorted array does not

C) Both sorted arrays and unsorted arrays have elements in ascending or descending order

D) None of the above

Q7 What is the difference between a for loop and a while loop in C language?

A) A for loop is executed a set number of times, a while loop is executed until a condition is met \*(Correct option)

B) A while loop is executed a set number of times, a for loop is executed until a condition is met

C) Both for loops and while loops are executed a set number of times

D) None of the above

Q8 What is the purpose of an if statement in C language?

A) To execute a block of code if a condition is met \*(Correct option)

B) To store data

C) To declare variables

D) None of the above

Q9 What is the purpose of the pointer to a pointer in C language?

A) To hold the address of another pointer \*(Correct option)

B) To hold the value of another pointer

C) To compare two pointers

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.How do you initialize a pointer in C?

a) By assigning the memory address of a variable to the pointer using the & operator \*(Correct option)

b) By assigning the value of a variable to the pointer

c) By using the malloc function

d) By using the calloc function

12.What is the difference between NULL and 0 in C?

a) NULL is a special value used to indicate that a pointer does not point to any valid memory location, while 0 is just a numerical value \*(Correct option)

b) NULL is used for arrays, while 0 is used for pointers

c) NULL is used for pointers, while 0 is used for arrays

d) There is no difference, both NULL and 0 can be used interchangeably

13.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+1)+1) + \*(\*(arr+2)+2));

a) 1

b) 14 \*(Correct option)

c) 15

d) 16

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

char name[10] = "Hello";

char \*ptr = name;

printf("%c", \*(ptr+2)+10);

a) 'H'

b) 'e'

c) 'v' \*(Correct option)

d) "llo"

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

char string1[70] = "CHITKARA";

char string2[70] = "chtkara";

int result;

result = strncasecmp(string1, string2, 3);

if (result == 0)

printf("Strings are equal.\n");

else if (result < 0)

printf("\"%s\" is less than \"%s\".\n", string1, string2);

else

printf("\"%s\" is greater than \"%s\".\n", string1, string2);

a) CHITKARA is less than chtkara.

b) "CHITKARA" is less than "chtkara". \*(Correct option)

c) CHITKARA is greater than chtkara.

d) "CHITKARA" is greater than "chtkara".

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

Q16. Chaitanya and Chitrang are best friends. Chaitanya was not sure about which algorithm to apply on a question given by their teacher. So, he was writing down each algorithm he tested till now on a paper separated by ‘0’ to indicate this didn’t work. Chaitanya and Chitrang used to call each algorithm by the number of characters present in it. Chitrang wants to help his friend. Hence He wants to know the current algorithm Chaitanya is applying. You have to help Chitrang as Chaitanya is very busy trying to get the solution.

**Input:**

a string

**Constraints:**

1<=**n**<=1000

**Output:**

**1** lines with a single integer giving the number of characters present in the last algorithm.

Sample test Cases

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| STC1 | twopointer0binarysearch0slidingwindow | 13 |
| STC2 | twopointer0binarysearch0hashing | 7 |

**Solution 16:**

#include<stdio.h>

int solve(char A[]) {

int n=0;

while(A[n] != '\0')

{

n++;

}

int i=n-1,c=0;

while(i>=0)

{

if(A[i] == '0')

{

break;

}

i--;

c++;

}

return c;

}

int main()

{

char a[1000];

scanf("%[^\n]s", a);

printf("%d",solve(a));

return 0;

}

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | abcd0hdgsdf0hsd0sdgfh | asdff | abc0abd0a |
| Output | 5 | 5 | 1 |

Q17. Write a function cyclicSwap that takes 3 numbers and swap them in cyclic order.

**Input:**

3 numbers

**Output:**

print numbers in cyclic order

Example -

Input-

|  |
| --- |
| Enter a, b and c respectively: 1  2  3  Output |
| Value before swapping:  a = 3  b = 1  c = 2  Value after swapping:  a = 2  b = 7  c = 4 |

**Solution**

#include <stdio.h>

void cyclicSwap(int \*a, int \*b, int \*c);

int main() {

int a, b, c;

printf("Enter a, b and c respectively: ");

scanf("%d %d %d", &a, &b, &c);

printf("Value before swapping:\n");

printf("a = %d \nb = %d \nc = %d\n", a, b, c);

cyclicSwap(&a, &b, &c);

printf("Value after swapping:\n");

printf("a = %d \nb = %d \nc = %d", a, b, c);

return 0;

}

void cyclicSwap(int \*n1, int \*n2, int \*n3) {

int temp;

// swapping in cyclic order

temp = \*n2;

\*n2 = \*n1;

\*n1 = \*n3;

\*n3 = temp;

}

**Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Case 1 | Test Case 2 | Test Case 3 |
| Input | Enter a, b and c respectively: 7  4  2 | Enter a, b and c respectively: 1  2  3 | Enter a, b and c respectively: 1  1  1 |
| Output | Value before swapping:  a = 7  b = 4  c = 2  Value after swapping:  a = 2  b = 7  c = 4 | Value before swapping:  a = 3  b = 1  c = 2  Value after swapping:  a = 2  b = 7  c = 4 | Value before swapping:  a = 1  b = 1  c = 1  Value after swapping:  a = 1  b = 1  c = 1 |

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

Q18 **Problem Statement:**  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.

Input

7

Output

20

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 9 | 8 | 4 |
| **Output** | 54 | 33 | 4 |

**#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);

}

}