Model Answer | Exam: SUM 2024

Subject: Programming In C | SUB CODE: 312303

Important Instructions to STUDENTS

1) *The model answer given here are prepared from the answers from the previously uploaded model answers by Board.*
2) *These model answers are not uploaded by the MSBTE official site but MSBTE study resources website prepared it for students. This model answer has question paper also inbuilt in it, no need to download it separate.*
3) *Please remembter that answers are not checked word to word but based on keywords which must be present in your answer*
4) *The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate*
5) *While assessing figures, examiner may give credit for principal components indicated in thefigure. The figures drawn bycandidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn*
6) *For programming language papers, credit may be given to any other program based on equivalentconcept*
7) *Students are advised to prepare all the syllabus from recommended book and use these model answers for the purpose of tests.*

Q.NO	SUB Q N	ANSWER	Marking Scheme
1.		Attempt any Five:-	10 Marks
	a)	List any four Keywords used in "C" ?	
	→	i) **int** :- To declare integer variables. ii) **float** :- To declare floating-point variables. iii) **Char** :- To declare character variables. iv) **return** :- To return a value from function.	
	b)	State any four math functions ?	
	→	i) **sqrt(x)** :- Computes square root of "x". ii) **sin(x)** :- Computes sin of "x" (in radians). iii) **abs(x)** :- Returns absolute value of "x". iv) **pow(x,y)** :- Computes "x" raised to power of "y".	

Q.NO	SUB Q N	ANSWER	Marking Scheme

1. C. Define Pointer. Write syntax for Pointer declaration?

→ **Pointer:-** Pointer is a variable that stores memory address of another variable.

It allows direct access to memory locations, enabling dynamic memory allocation & efficient handling of arrays & structures.

```
int *ptr; // Declares a pointer to an integer.
```

d. Write the syntax of switch case statement?

→ Switch case statement is used for
- multi-way decision making.

```
switch (expression) {
    case constant 1:
        // statements
        break;
    case constant 2:
        // statements
        break;
    default:
        // statements
```

Example:-

```
int num = 2;
switch (num) {
    case 1: printf("One"); break;
    case 2: printf("Two"); break;
    default: printf("Invalid");
```

Q.NO	SUB Q N	ANSWER	Marking Scheme

1. e. Define array. List its types?

→ **Array :-** An array is a collection of similar data items stored in contiguous memory locations.

Types of arrays :-

1] One - dimensional array.
(linear collection of elements).

Ex:- `int arr[5]={1,2,3,4,5};`

2] Two - dimensional array.
(matrix like collection of elements).

Ex:- `int arr[2][3]={{1,2,3},{4,5,6}};`

3] Multi - dimensional array.
(array with more than two dimensions).

Ex:- `int arr[2][3][4];`

f] Define type casting. Give one example?

→ **Type Casting :-** It is the process of converting one data type into another.

ex:-
```
int a=10;
float b=(float)a; // Explit type casting
                  from int to float.
```

or] Declare a structure student with elements roll_no & name?

→ A structure is a user defined data type that groups related data items of different data types.

ex:-
```
struct student {
    int roll_no; // stores roll number
  char name[50]; // stores name
};
```

Q.NO	SUB Q N	ANSWER	Marking Scheme
2.		Attempt any Three.	12 Marks
	a]	Develop a 'C' program for addition & multiplication of two integer numbers?	
	→		

```
#include <stdio.h>
int main(){
    int a, b, sum, mul;
    printf("Enter two integers:");
    scanf("%d %d", &a, &b); //input two integers
    sum = a + b; // Calculate sum
    mul = a * b; // Calculate multiplication
    printf("Sum: %d\n", sum); //Output sum
    printf("Multiplication: %d\n", mul); //Output multiplication
    return 0;
}
```

Explanation:
- program takes two integers as input from user.
- It calculates sum & multiplication of two numbers.
- finally prints the results.
- program demonstrates basic arithmetic operations in C.

Q.NO	SUB Q N	ANSWER	Marking Scheme

2. b. Explain pointer arithmatic with an example.

→ Pointer arithmatic = involves performing arithmatic operations like addition, substraction on pointers.

When you add an integer to a pointer, it moves pointer forward by that many elements of the type it points to.

```
# include < stdio.h >
int main ( ){
  int arr [ ] = { 10, 20, 30, 40, 50 };
  int *ptr = arr ; // ptr points to first element of arr
  printf ("initial value: %d \n", *ptr); //Output : 10
  ptr ++; // Move pointer to next element
  printf ("After increment: %d \n", *ptr); //Output: 20
  ptr + = 2; // Move pointer by 2 elements
  printf ("After adding 2: %d \n", *ptr); //Outpt : 40
  return 0;
}
```

Explanation :-

- pointer ptr initially points to first element of array arr.
- After incrementing (Ptr ++), it points to second element.
- Adding 2 (ptr + = 2) moves pointer to fourth element.
- This demonstrates how pointer arithmetic works in C.

Q.NO	SUB Q N	ANSWER	Marking Scheme

2. c. Explain following functions:- (i) getchar(), (ii) putchar(), (iii) getch(), (iv) putch(). ?

→ (i) getchar():- Reads single character from standard input(Keyboard)

```
char ch = getchar();//
```

(ii) putchar():- Writes single character to standard output(console).

```
putchar('A');//
```

(iii) getch():-

Reads single character from keyboard without echoing it to the screen. (without displaying it)

```
char ch = getch();//
```

(iv) putch():- Writes a single character to the console.

```
putch('B');//
```

d. Write a program to print Fibonacci series?

→

```
#include <stdio.h>
int main(){
    int n,a=1,b=1,c;
    printf("Enter number of terms;");
    scanf("%d"&n);// input number of terms
    printf("Fibonacci Series; %d %d",a,b);// Print
                                    first two terms
    for(int i=2 ; i<n; i++){
        c=a+b;// calculate next term
        printf("%d",c);// Print next term
        a=b; // update a
        b=c; // update b
    }
    return 0;
}
```

Explanation - fibonacci series starts with 1,1 & each subsequent term is sum of previous two terms.
Program calculates & prints fibonacci series upto n terms.
It demonstrates use of loops & basic arithmatic in "C".

Q.NO	SUB Q N	ANSWER	Marking Scheme
3.		Attempt any three.	12 Marks

a] Explain conditional operator with an example?

→ Conditional operator (?:);

It is a ternary operator that works like an if else statement.

```
Condition ? expression1 : expression2;
```

if, condition is true, expression1 is evaluated, otherwise expression2 is evaluated.

```
# include < stdio.h >
int main( ) {
    int a = 10 , b = 20;
    int max = ( a > b) ? a : b ; // max will be 20
    printf ("Max: %d\n ", max);
    return 0 ;
}
```

Explaination - Program compares two numbers a & b if a is greater than b max is assigned value of a, otherwise it is assigned value of b.

It demonstrates use of conditional operator for decision making.

b] Explain any two string functions with examples.?

→ i) strlen(): Returns length of string.

(number of characters excluding null character \0)

```
# include < stdio.h >
# include < string.h >
int main( ){
    char str[ ] = "Hello";
    int len = strlen (str); // len will be 5.
    printf ("Length : % d\n", len);
    return 0;
}
```

(8)

Q.NO	SUB Q N	ANSWER	Marking Scheme

3 ii) strcpy(): Copies one string to another.

```
#include <stdio.h>
#include <string.h>
int main(){
        char src[] = "Hello";
        char dest[10];
        strcpy(dest, src); //dest will contain "Hello"
        printf("Destination: %s\n", dest);
        return 0;
}
```

c] Differenciate between while & do while loop?

→ i] While loop: checks condition before executing the loop body.

If condition false initially, loop body will not execute at all.

```
while (condition) {
    // statements.
```

ii] do-while loop: Excutes loop body at least once before checking the condition.

Even if condition is false initially, loop body will execute once.

```
do {
    // statements
} while (condition);
```

Q.NO	SUB Q N	ANSWER	Marking Scheme

3 d] Write a program to accept 5 data elements in an array & arrange them in asceding order?

→

```
#include <stdio.h>
int main(){
    int arr[5], i, j, temp;
    printf("Enter 5 elements:");
    for(i=0; i<5; i++){
        scanf("%d", &arr[i]; //Input 5 elements
    }
    for(i=0; i<5; i++){
        for(j=i+1; j<5; j++){
            if(arr[i] > arr[j]){ //Swap if out of order.
                temp = arr[i];
                arr[i]= arr[j];
                arr[j]= temp;
            }
        }
    }
    printf("Sorted array:");
    for(i=0; i<5; i++){
        printf("%d", arr[i]; //Output sorted arry.
    }
    return 0;
}
```

Explaination:-

Program uses bubble sort algorithm to sort array in ascending order.

It compaires each element with next one & swaps them if they are out of order.

It demonstrates use of nested loops & basic sorting in "C".

(10)

Q.NO	SUB Q N	ANSWER	Marking Scheme
4.		Attempt any three.	12 Marks
	a]	Describe general structure of a 'C' Program. ?	
	→	**C program** consists of following parts:-	
		i] Preprocessor Directives :- include header files. (eg. # include < stdio.h >).	
		ii] Global Declarations:- Declare global variables & functions.	
		iii] Main function :- Entry point of program. [int main ( )]	
		iv] Local Declarations:- Declare variables inside functions.	
		v] Statements :- Executable code.	
		vi] Return statement:- Indicates end of function (return 0;)	

11

Q.NO	SUB Q N	ANSWER	Marking Scheme

4. b. Differenciate between call by value & call by referance?

→ i] Call by value:-

A of actual value is passed to the function. Changes made to parameter inside function do not affect original value.

```
void increment(int a) {
    a++;
}
int main() {
    int num = 10;
    increment(num); // num remains 10.
    return 0;
}
```

ii] Call by Referance:

Address of actual value passed to function. Changes made to parameter inside function affect the original value.

```
void increment(int *a) {
    (*a)++;
}
int main() {
    int num = 10;
    increment(&num); // num becomes 11.
    return 0;
}
```

Q.NO	SUB Q N	ANSWER	Marking Scheme

4. c] Write a program to print sum of digits in given number?

→

```
#include<stdio.h>
int main(){
    int num, sum = 0, rem;
    printf("Enter a number:");
    scanf("%d", &num); // Input number
    while(num!=0){
        rem = num % 10; // Extract last digit
        sum += rem; // Add to sum
        num /= 10; // Remove last digit.
    }
    printf("Sum of digits: %d\n", sum); // Output sum.
    return 0;
}
```

Explaination-
Program extracts each digit of number using modulo operator(%) & adds it to the sum.
It demonstrates use of loops & arithmetic operations in C.

d] Write program to demonstrate concept of pointer to a function?

→

```
#include<stdio.h>
void display(){
    printf("Hello, World!\n");
}
int main(){
    void(*func_ptr)() = display; // Pointer to function
    func_ptr(); // Call function using pointer.
    return 0;
}
```

Explaination:- Program declares a pointer to a function (func_ptr) & assigns it address of display function.
It then calls function using pointer.
It demonstrates how function pointer work in "C".

Q.NO	SUB Q N	ANSWER	Marking Scheme
5.		Attempt any two?	12 Marks

→ a) Write program using switch statement to check whether entered character is a vowel or consonant?

→

```
#include<stdio.h>
int main(){
    char ch;
    printf("Enter a character");
    scanf("%c",&ch);//
    switch(ch){
      case'a':case'e':case'i':case'o':case'u':
      case'A':case'E':case'I':case'O':case'U':
      printf("Vowel\n");//Output if vowel
      break;
      default:
          printf("Consonant\n");//Output if consonant.
    }
    return 0;
}
```

Explaination –

program uses a switch statement to check if entered character is vowel or consonant.

It covers both lowercase & uppercase vowels.

It demonstrates statement for decision making.

Q.NO	SUB Q N	ANSWER	Marking Scheme

5. b. Write a program for addition of two 3X3 matrices?

→

```
#include <stdio.h>
int main(){
    int mat1[3][3], mat2[3][3], sum[3][3], i, j;
    printf("Enter elements of first matrix:\n");
    for(i=0; i<3; i++){
      for(j=0; j<3; j++){
        scanf("%d", &mat1[i][j]); // Input first matrix.
      }
    }
    printf("Enter elements of second matrix:\n");
    for(i=0; i<3; i++){
      for(j=0; j<3; j++){
        scanf("%d", &mat2[i][j]); // Input second matrix.
      }
    }
    for(i=0; i<3; i++){
      for(j=0; j<3; j++){
        sum[i][j] = mat1[i][j] + mat2[i][j]; // Calculate sum.
      }
    }
    printf("Sum of matrices:\n");
    for(i=0; i<3; i++){
      for(j=0; j<3; j++){
        printf("%d", sum[i][j]); // Output sum matrix.
      }
      printf("\n");
    }
    return 0;
}
```

Explaination - program takes two 3X3 matrices as input & calculate their sum.
It uses nested loops to iterate through matrices & perform the addition.

It Demonstrates how to work with 2D arrays & perform matrix operations in "C."

Q.NO	SUB Q N	ANSWER	Marking Scheme

5. c. Write a program to find factorial of a number using recursion ?

→

```
#include <stdio.h>
int factorial(int n){
    if(n==0) return 1; // Base case.
    return n*factorial(n-1); // Recursive call.
}
int main(){
    int num;
    printf("Enter a number:");
    scanf("%d",&num); // Input number
    printf("Factorial: %d\n",factorial(num)); // Output factorial
    return 0;
}
```

Explaination:-

Program calculates factorial of number using recursion
base case is when n==0, which returns 1.
recursive case multiplies n by factorial of n-1.
It demonstrates use of recursion in 'C'.

Q.NO	SUB Q N	ANSWER	Marking Scheme
6.		Attempt any two.	12 Marks
	a]	Differenciate between break & continue statements. State use of break & continous statements with examples.	

→ i] break:- Exits loop or switch statement immediately.

```
for(int i=0; i<10; i++){
    if(i==5) break; // loop exits when i is 5.
}
```

ii] Continue:- Skips the current iteration & continues with next iteration of the loop.

```
for(int i=0; i<10; i++){
    if(i==5) continue; // skips iteration when i is 5.
}
```

Q.NO	SUB Q N	ANSWER	Marking Scheme

6. b. Write a program to declare a structure employee having data members name, age, street & city. Accept data for two employees & display it.?

→

```
#include<stdio.h>
    struct employee{
        char name[50];
        int age;
        char street[50];
        char city[50];
    }
    int main(){
      struct employee emp[2];
      for(int i=0;i<2;i++){
        printf("Enter details for employee %d:\n",i+1);
        printf("Name:"); scanf("%s",emp[i].name);
        printf("Age:"); scanf("%d",emp[i].age);
        printf("Street:"); scanf("%s",emp[i].street);
        printf("City:"); scanf("%s",emp[i].city);
      }
      for(int i=0;i<2;i++){
        printf("\n Employee %d:\n",i+1);
        printf("Name:%s\n",emp[i].name);
        printf("Age:%d\n",emp[i].age);
        printf("street:%s\n",emp[i].street);
        printf("City:%s\n",emp[i].city);
      }
      return 0;
    }
```

Explaination:- Program declares a structure employee with four data, members: name, age, street & city. It accepts data for two employees & displays the details. It demonstrates use of structures in "C".

Q.NO	SUB Q N	ANSWER	Marking Scheme

6. C. Explain with example array of pointers ?

→ An array of pointers is an array where each element is a pointer.

```
int a=10, b=20, c=30;
int *arr[3]={&a, &b, &c}; // Array of pointers.
for(int i=0; i<3; i++){
    printf("%d",*arr[i]); // Output:10 20 30
}
```

Explanation:-

Program declares an array of pointers to integers. Each element of array points to a different integer variable.

It demonstrates how to use arrays of pointers in 'C'.