

# CSA02 C Progrmming

N PAVAN SAI

192225023

The screenshot shows the SIMATS C IDE interface. The top bar displays the SIMATS logo and 'Saveetha School of Engineering'. The user's name 'N PAVAN SAI' and ID '192225023' are in the top right. The left panel contains the question text for CHQ1: 'A Pythagorean Triplet is a set of three integers m, n and o such that  $m^2+n^2=o^2$ . Given a limit A, generate all Pythagorean triples with value smaller than given limit A'. It includes sample input 'A=12' and sample output '(3,4,5), (6,8,10)'. The right panel shows test cases: 1. A=10, 2. A=-10, 3. A=#, 4. A=1.5, 5. A=25. The main editor shows a C program that takes two numbers as input and checks if they form a Pythagorean triplet. The output area shows the input '3' and '4', and the message 'The given number is pythagoren triple'.

```
1. #include <stdio.h>
2. #include <conio.h>
3. #include <math.h>
4. int main()
5. {
6.     int a,b,c;
7.     printf("Enter a number :");
8.     scanf("%d",&a);
9.     printf("Enter a number");
10.    scanf("%d",&b);
11.    if (c=sqrt(a*a+b*b))
12.    {
13.        printf("The given number is pythagoren triple");
14.    }
15.    else
16.        printf("The given number is not a pythagoren triple");
17.    return 0;
18. }
```

The screenshot shows the SIMATS C IDE interface for a bubble sort problem. The top bar is the same as the previous screenshot. The left panel contains the question text for CHQ9: 'Bubble sort is a straightforward sorting algorithm that checks and swaps elements if they are not in the intended order. It compares two adjacent elements to find which one is greater or lesser and switches them based on the given condition until the final place of the element is found. Write a Program to sort the given numbers in an array using bubble sort'. It includes sample input 'Enter the number of Elements: 5' and 'Enter the Elements: -2 45 0 11 -9'. The right panel shows test cases: CHQ19, CHQ2, CHQ20, CHQ3, CHQ4, CHQ5, CHQ6, CHQ7, CHQ8, CHQ9. The main editor shows a C program that implements bubble sort. The output area shows the input array '-2 45 0 11 -9' and the sorted array '-9 -2 0 11 45'.

```
1. #include <stdio.h>
2. void swap(int* xp, int* yp)
3. {
4.     int temp = *xp;
5.     *xp = *yp;
6.     *yp = temp;
7. }
8. void bubblesort(int arr[], int n)
9. {
10.    int i, j;
11.    for(i = 0; i < n - 1; i++)
12.        for (j = 0; j < n - i - 1; j++)
13.            if (arr[j] > arr[j + 1])
14.                swap(&arr[j], &arr[j + 1]);
15. }
16. void printarray(int arr[], int size)
17. {
18.     int i;
19.     for (i = 0; i < size; i++)
20.         printf("%d ", arr[i]);
21.     printf("\n");
22. }
23. int main()
24. {
25.     int arr[] = {-2,45,0,11,-9};
26.     int n = sizeof(arr) / sizeof(arr[0]);
27.     bubblesort(arr, n);
28.     printf("sorted array: \n");
29.     printarray(arr, n);
30.     return 0;
31. }
```

Questions  
CHQ16.

Write a C programming to find out maximum and minimum of some values using function which will return an array.

Test Data :  
Input 5 values  
25  
11  
35  
65  
20

Expected Output :  
Minimum value is : 11

## Test Cases

CHQ11  
CHQ10  
CHQ11  
CHQ12  
CHQ13  
CHQ14  
CHQ15  
CHQ16  
CHQ17  
CHQ18

```
1. #include<stdio.h>
2. void findminmax(int arr[], int size, int*min,int*max)
3. {
4.     *min=arr[0];
5.     *max=arr[0];
6.     for (int i = 1; i<size; i++){
7.         if(arr[i]<*min)
8.             *min = arr[i];
9.         else if (arr[i] > *max)
10.            *max = arr[i];
11.     }
12. }
13.
14.
15.
16. int main(){
17.     int arr[] = { 25,11,35,65,20};
18.     int size = sizeof(arr) / sizeof(arr[0]);
19.     int min,max;
20.     findminmax(arr,size,&min,&max);
21.     printf("min valueis :%d\n",min);
22.     printf("max value is : %d\n",max);
23.     return 0;
24. }
```

Your Input Goes Here....!!!

min valueis :11  
max value is : 65

Questions  
CHQ5.

Write a program in C to check Armstrong and perfect numbers using the function.

Test Data :  
Input any number: 371  
Expected Output :  
The 371 is an armstrong number.  
The 371 is not a Perfect number.

## Test Cases

CHQ19  
CHQ2  
CHQ20  
CHQ3  
CHQ4  
CHQ5  
CHQ6  
CHQ7  
CHQ8  
CHQ9

```
1. #include<stdio.h>
2. int isarmstrong(int n){
3.     int sum=0,temp=n,digit;
4.     while(temp!=0){
5.         digit=temp%10;
6.         sum+=(digit*digit*digit);
7.         temp/=10;
8.     }
9.     return sum==n;
10. }
11. int isperfect(int n){
12.     int sum=0,i;
13.     for(i=1;i<=n/2;i++){
14.         if(n%i==0){
15.             sum+=i;
16.         }
17.     }
18.     return sum==n;
19. }
20. int main(){
21.     int n;
22.     printf("Enter a number:");
23.     scanf("%d",&n);
24.     if(isarmstrong(n)){
25.         printf("\n%d is armstrong number",n);
26.     }else{
27.         printf("\n%d is not armstrong number",n);
28.     }
29.     if(isperfect(n)){
30.         printf("\n%d is perfect number",n);
31.     }else{
32.         printf("\n%d is not perfect number",n);
33.     }
34.     return 0;
35. }
```

371

Enter a number:  
371 is armstrong number  
371 is not perfect number

Questions  
CHQ2.

Ask the user to enter any positive integer. Compile and Execute the C program to check whether

Sample Input:

Enter the number to check for strong number: 145

Sample Output:

Factorial of 5 is =120

Factorial of 4 is =24

Factorial of 1 is =1

Sum of the factorial of the a given number i145 is =145

145 is the strong number.

## Test Cases

1. 153
2. 26
3. 40585
4. 160
5. 59.5

CHQ19

CHQ20

CHQ3

CHQ4

CHQ20

CHQ4

CHQ2

CHQ3

CHQ2

C

Run

Save

Logout

```
1. #include<stdio.h>
2. int main()
3. {
4.     int i, originalnum, num, lastdigit, sum;
5.     long fact;
6.     printf("Enter any number to check strong number:");
7.     scanf("%d", &num);
8.     originalnum=num;
9.     sum=0;
10.    while(num>0)
11.    {
12.        lastdigit=num%10;
13.        fact=1;
14.        for(i=1; i<=lastdigit;i++)
15.        {
16.            fact=fact*i;
17.        }
18.        sum=sum+fact;
19.        num=num/10;
20.    }
21.    if(sum==originalnum)
22.    {
23.        printf("%d is strong number",originalnum);
24.    }
25.    else
26.    {
27.        printf("%d is not strong number",originalnum);
28.    }
29.    return 0;
30. }
```

145

enter any number to check strong number:145 is strong number

Questions  
CHQ7.

Write a program to search the given element using binary search method and display its position in a linear array.

Sample Input:

Array of elements = {16, 18, 27, 16, 23, 21, 19}

Element to search = 23

Sample Output:

Given element 23 is found at 5 th position

## Test Cases

CHQ19

CHQ20

CHQ3

CHQ4

CHQ20

CHQ4

CHQ2

CHQ3

CHQ2

C

Run

Save

Logout

```
1. #include<stdio.h>
2. int binarysearch(int arr[],int i,int r,int x);
3. int main(){
4.     int arr[]={16,18,27,16,23,21,19};
5.     int n=sizeof(arr)/sizeof(arr[0]);
6.     int x=23;
7.     int result=binarysearch(arr,0,n-1,x);
8.     if(result==1)
9.     {printf("Element not found\n");}
10.    else
11.    {printf("Element found at position %d in array\n",result);}
12.    }
13.    return 0;
14. }
15. int binarysearch(int arr[],int i,int r,int x){
16.     while(i<=r){
17.         int mid=i+(r-i)/2;
18.         if(arr[mid]==x){
19.             return mid;
20.         }
21.         else if(arr[mid]<x){
22.             i=mid+1;
23.         }
24.         else{
25.             r=mid-1;
26.         }
27.     }
28.     return -1;
29. }
```

Your Input Goes Here...!!!

Element found at position -2 in array

## Questions

CHQ11.

Write a C program for Inserting an element to the array at the specified index.

Sample Input:

Enter the number of Elements: 5

Enter the Elements:

47 34 21 89 12

Element to be inserted 100

At position: 4

Output:

12 21 34 100 89 12

## Test Cases

CHQ11

CHQ10

CHQ11

CHQ12

CHQ13

CHQ14

CHQ15

CHQ16

CHQ17

CHQ18

```
1. #include<stdio.h>
2. int main()
3. {
4.     int a[10],i,n,ele,pos;
5.     printf("Enter no.of elements:");
6.     scanf("%d",&n);
7.     printf("Elements in array:\n");
8.     for(i=1;i<=n;i++)
9.     {
10.         scanf("%d",&a[i]);
11.         printf("Elements are:\n");
12.         for(i=1;i<=n;i++)
13.         {
14.             printf("%d\t",a[i]);
15.         }
16.     }
17.     printf("Enter the position:\n");
18.     scanf("%d",&pos);
19.     printf("Enter element u want to enter:\n");
20.     scanf("%d",&ele);
21.     for(i=1;i<=n;i++)
22.     {
23.         if(i>pos)
24.             a[i]=a[i-1];
25.         else
26.         {
27.             if(i==pos)
28.                 a[i]=ele;
29.         }
30.     }
31.     printf("Array after inserting:\n");
32.     for(i=1;i<=n;i++)
33.     {
34.         scanf("%d\t",a[i]);
35.     }
36. }
```

5

47

34

21

89

12

Runtime Error

## Questions

CHQ10.

The selection Sort is a sorting algorithm that works by finding the smallest number from the array and then placing it to the first position. The next array that is to be traversed will start from index next to the position where the smallest number is placed. Write a Program to sort the given numbers in an array using selection sort.

Sample Input:

Enter the number of Elements: 5

Enter the Elements:

47 34 21 89 12

Output:

## Test Cases

CHQ10

CHQ10

CHQ11

CHQ12

CHQ13

CHQ14

CHQ15

CHQ16

CHQ17

CHQ18

```
1. #include<stdio.h>
2. int main()
3. {
4.     int arr[5]={47,34,21,89,12};
5.     int n=5;
6.     int i,j,position,swap;
7.     for(i = 0; i < (n - 1); i++)
8.     {
9.         position=i;
10.         for(j = i + 1; j < n; j++)
11.         {
12.             if(arr[position]>arr[j])
13.                 position=j;
14.         }
15.         if(position !=i)
16.         {
17.             swap=arr[i];
18.             arr[i]=arr[position];
19.             arr[position] = swap;
20.         }
21.     }
22.     for(i=0;i<n;i++)
23.         printf("%d\t",arr[i]);
24.     return 0;
25. }
```

5

47 34 21 89 12

0 9 9 9 9 9 9 10  
9

Questions  
CHQ16.

Write a program in C to find the sum of the series  $1/1+2!/2+3!/3+4!/4+5!/5$  using the function

Expected Output :

The sum of the series is : 34

## Test Cases

CHQ16  
CHQ10  
CHQ11  
CHQ12  
CHQ13  
CHQ14  
CHQ15  
CHQ16  
CHQ17  
CHQ18

C

Run

Save

Logout

```
1. #include<stdio.h>
2. int fact(int);
3. void main()
4. {
5.     int sum;
6.     sum=fact(1)/1+fact(2)/2+fact(3)/3+fact(4)/4+fact(5)/5;
7.     printf("\n\n function:find the sum of 1!/1+2!/2+3!/3+4!/4+5!/5:\n");
8.     printf("the sum of the series is:%d\n\n",sum);
9. }
10. int fact(int n)
11. {
12.     int num=0,f=1;
13.     while(num<=n-1)
14.     {
15.         f=f*f*num;
16.         num++;
17.     }
18.     return f;
19. }
```

Your Input Goes Here...!!!

function:find the sum of  
1!/1+2!/2+3!/3+4!/4+5!/5:  
the sum of the series is:34