

Q1) Design a program to create a smart counter that keeps track of the number of times a function is called. The program should utilize the auto keyword for type inference and the static keyword for maintaining the state of the counter across function calls.

#### Requirements

Implement a function smartCounter() that doesn't take any arguments and returns an integer.

Each time smartCounter() is called, it should increment a static counter variable by 1 and return the updated count.

The counter should retain its value between function calls.

#### Tasks:

Implement the smartCounter() function with appropriate use of the static keyword to maintain the counter state.

Write a main program to test the functionality of smartCounter().

Test the program with multiple function calls to ensure the counter increments correctly.

```
rps@rps-virtual-machine: ~/Desktop/charan/day3
1 #include<stdio.h>
2 int smartcounter()
3 {
4     static int count=0;
5     count++;
6     return count;
7 }
8 int main(){
9     printf("count :%d\n",smartcounter());
10    printf("count :%d\n",smartcounter());
11    printf("count :%d\n",smartcounter());
12    printf("count :%d\n",smartcounter());
13
14    return 0;
15 }
~
~
~
~
~
~
"smartcounter.c" 15L, 270B
```

## OUTPUT

```
rps@rps-virtual-machine: ~/Desktop/charan/day3
rps@rps-virtual-machine:~/Desktop$ cd charan
rps@rps-virtual-machine:~/Desktop/charan$ cd day3
rps@rps-virtual-machine:~/Desktop/charan/day3$ vim smartcounter.c
rps@rps-virtual-machine:~/Desktop/charan/day3$ gcc smartcounter.c
rps@rps-virtual-machine:~/Desktop/charan/day3$ ./a.out
count :1
count :2
count :3
count :4
rps@rps-virtual-machine:~/Desktop/charan/day3$
```

2.write a program to find a factorial of a number without using function.

```
rps@rps-virtual-machine: ~/Desktop/charan/day3

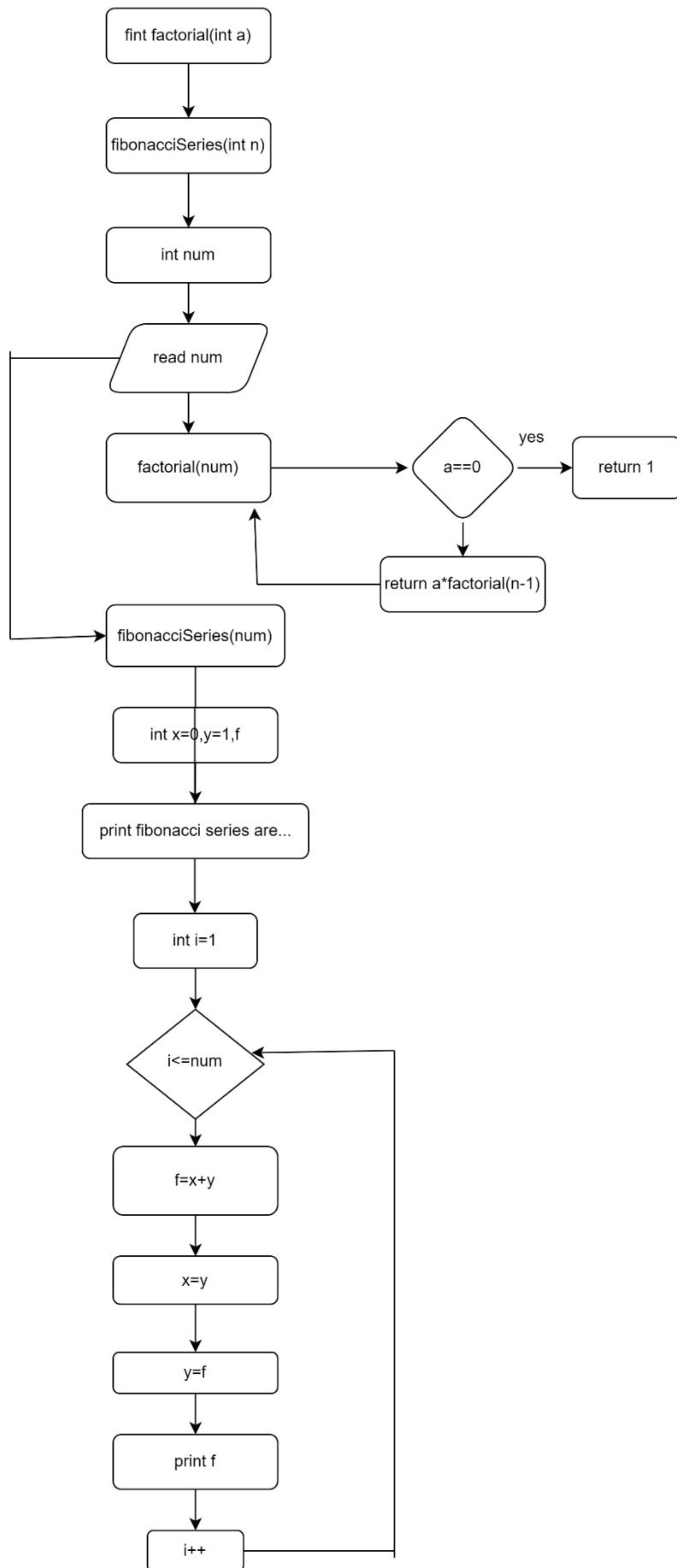
1 #include<stdio.h>
2 int factorial(int a);
3 void fibonacciSeries(int s);
4 int main()
5 {
6     int num,val;
7     scanf("%d",&num);
8     int b=factorial(num);
9     printf("The factorail of num are %d\n",b);
10    fibonacciSeries(num);
11 }
12 int factorial(int a)
13 {
14     if(a==0)
15     {
16         return 1;
17     }
18     else
19     {
20         return a*factorial(a-1);
21     }
22 }
23 void fibonacciSeries(int s)
24 {
25     int x=0;
26     int y=1,f;
27     printf("The fibonacci series are ....\n");
28     for(int i=1;i<=s;i++)
29     {
30         f=x+y;
31         x=y;
32         y=f;
33         printf("%d\n",f);
34     }
35 }
```

OUTPUT

```
rps@rps-virtual-machine: ~/Desktop/charan/day3

rps@rps-virtual-machine:~/Desktop/charan/day3$ gcc fact.c
rps@rps-virtual-machine:~/Desktop/charan/day3$ ./a.out
6
The factorail of num are 720
The fibonacci series are ....
1
2
3
5
8
13
rps@rps-virtual-machine:~/Desktop/charan/day3$
```

FLOWCHART



3)write a program to create a character array of size 10 and input characters from a user and count the occurrence of characters in the array (Participant must write tracing of this program)

```
rps@rps-virtual-machine: ~/Desktop/charan/day3/strings
1 #include<stdio.h>
2 int main()
3 {
4     char arr[10];
5     int count=0;
6     printf("enter the string in lows caps...");
7     scanf("%s",arr);
8     for(int i=0;i<10;i++)
9     {
10         if(arr[i]=='a' ||arr[i]=='b' || arr[i]=='c' || arr[i]=='d' ||arr[i]=='e' || arr[
i]=='f' || arr[i]=='g' || arr[i]=='h' || arr[i]=='i' || arr[i]=='j' || arr[i]=='k' || arr[i]=='
l' || arr[i]=='m' || arr[i]=='n' || arr[i]=='o' || arr[i]=='p' || arr[i]=='q' || arr[i]=='r' ||
arr[i]=='s' || arr[i]=='t' || arr[i]=='u' || arr[i]=='v' || arr[i]=='x' || arr[i]=='y' || arr[
i]=='z')
11         {
12             count++;
13         }
14     }
15     printf("Count of characters in character Array are %d\n",count);
16 }
~
~
~
~
"charArray.c" 16L, 615B 7,2-9 All
```

## OUTPUT

```
rps@rps-virtual-machine: ~/Desktop/charan/day3/strings
rps@rps-virtual-machine:~/Desktop/charan/day3/strings$ vim charArray.c
rps@rps-virtual-machine:~/Desktop/charan/day3/strings$ gcc charArray.c
rps@rps-virtual-machine:~/Desktop/charan/day3/strings$ ./a.out
enter the string in lows caps...charan
Count of characters in character Array are 6
rps@rps-virtual-machine:~/Desktop/charan/day3/strings$
```

4) Write a C program that performs various operations as follows:

(Initialize two strings str1 and str2 with some values as follows str1 Welcome\_to and str2 Digital\_Era )

Concatenation:

Concatenate the strings

Copying:

Copy the concatenated string to another string and print the copied string.

Comparison:

Compare the original concatenated string with the copied string and print whether they are equal, or which one is lexicographically greater.

Length:

Find and print the length of both strings.

Search:

Find the first occurrence of the character 'o' in the original concatenated string and print its position. and manipulations to the console.

The participant can use string.h to perform the above functions

```
rps@rps-virtual-machine: ~/Desktop/charan/day3/strings
1 #include<stdio.h>
2 #include<string.h>
3 int main()
4 {
5     char str1[]="Welcome_to ",str2[]=" Digital_Er\n";
6     strcat(str1,str2);
7     printf("%s",str1);
8     for(int i=0;i<strlen(str1);i++)
9     {
10         if(str1[i]=='o')
11         {
12             printf("%d\n",i);
13             break;
14         }
15     }
16
17     char str3[50];
18     strcpy(str3,str2);
19     printf("%s\n",str3);
20     //comparison
21     int cmp=strcmp(str1,str2);
22     if(cmp>0)
23     {
24         printf("%s is greater than %s\n",str1,str2);
25     }
26     else if(cmp<0)
27     {
28         printf("%s is greater than %s\n",str2,str1);
29     }
30     else
31     {
32         printf("%s is equal to %s\n",str1,str2);
33     }
34     printf("The length of str1 is %zu\n",strlen(str1));
35     printf("The length of str2 is %zu\n",strlen(str2));
36
37     return 0;
38 }
39
```

OUTPUT

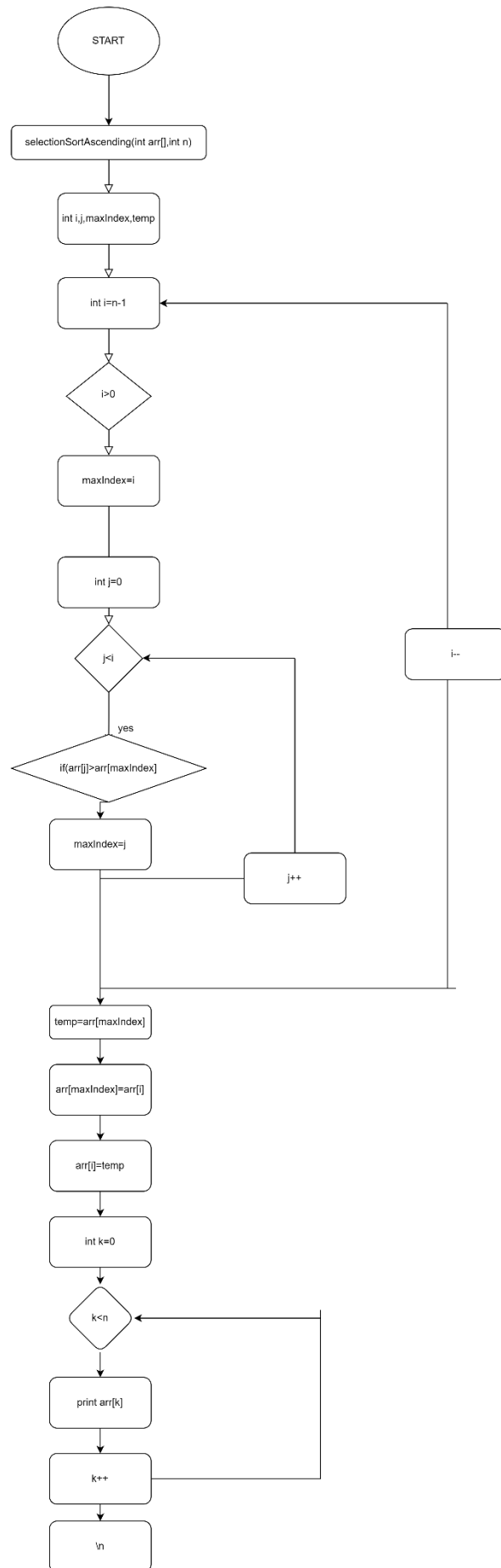
```
rps@rps-virtual-machine: ~/Desktop/charan/day3/strings
rps@rps-virtual-machine:~/Desktop/charan/day3/strings$ gcc ques4.c
rps@rps-virtual-machine:~/Desktop/charan/day3/strings$ ./a.out
Welcome_to  Digital_Er
4
Digital_Er

Welcome_to  Digital_Er
is greator than Digital_Er

The length of  str1 is 23
The length of str2 is 11
rps@rps-virtual-machine:~/Desktop/charan/day3/strings$
```

5) Pseudocode and Flowchart for Sorting Algorithm - Write pseudocode and create a flowchart for a bubble sort algorithm. Provide a brief explanation of how the algorithm works and a simple array of integers to demonstrate a dry run of your algorithm. (use a lucid chart for flowchart and perform a dry run on the following sample array: 300,500,600,45,65,15,50)

SELECTION SORT:





## BUBBLE SORT

