

CSA0672 SLOT D –DAA

S.MANIKANTA

REG NO 192011145

DAY 1[6-15]

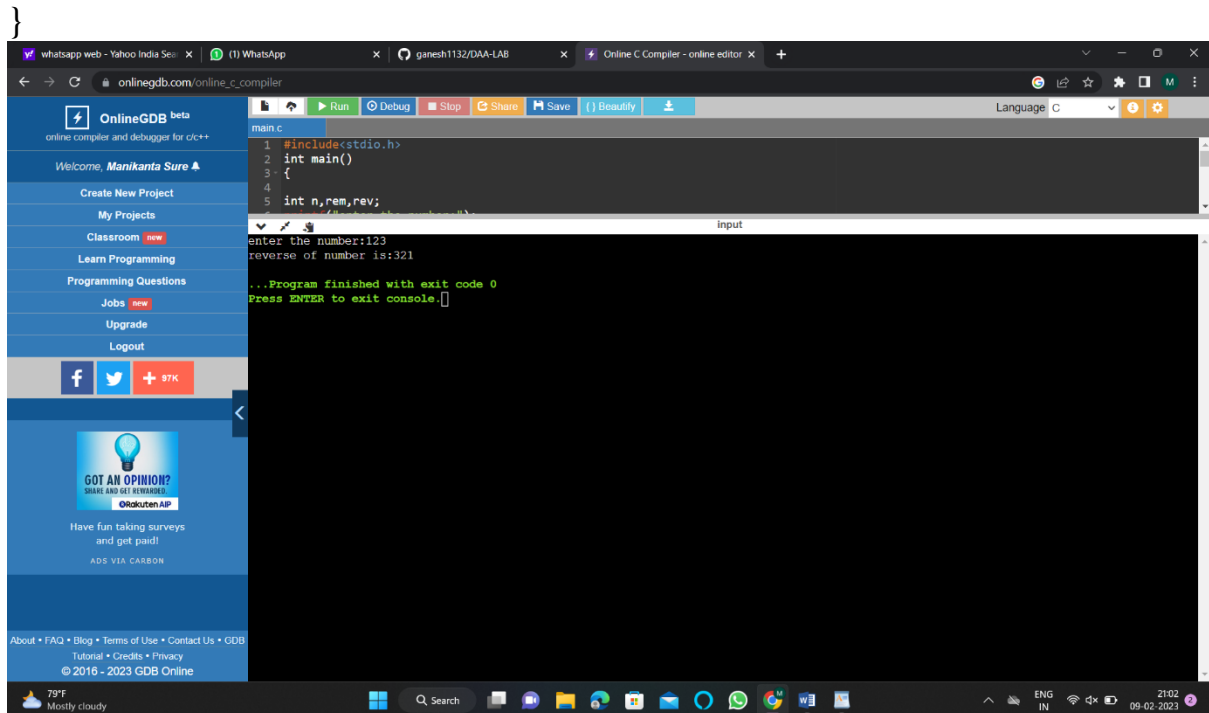
Q.7 Write a program to generate all the reverse of a prime should be prime

program

```
#include<stdio.h>

int main()
{

    int n,rem,rev;
    printf("enter the number:");
    scanf("%d",&n);
    while(n!=0)
    {
        rem=n%10;
        rev=rev*10+rem;
        n=n/10;
    }
    printf("reverse of number is:%d",rev);
    return 0;
```



Q.8 Compute the program to find the GCD of two numbers. And also find the time of Recursion

Program

```
#include <stdio.h>

int main()
{
    int n1, n2, i, GCD_Num;
    int count=0;
    printf ( " Enter any two numbers: \n ");
    scanf ( "%d %d", &n1, &n2);

    for( i = 1; i <= n1 && i <= n2; ++i)
    {
        count++;
        if (n1 % i ==0 && n2 % i == 0)
```

```

        GCD_Num = i;
        count++;
    }

    count++;

    printf("gcd of two numbers %d and %d is %d ", n1, n2, GCD_Num);
    printf("time complexity :%d ",count);
    return 0;
}

```

The screenshot shows a web browser with the OnlineGDB compiler. The code in the editor is as follows:

```

1 #include <stdio.h>
2 int main()
3 {
4     int n1, n2, i, GCD_Num;
5     int count=0;
6     while(n1!=n2)
7     {
8         if(n1>n2)
9             n1=n1-n2;
10        else
11            n2=n2-n1;
12        GCD_Num = i;
13        count++;
14    }
15    count++;
16    printf("gcd of two numbers %d and %d is %d ", n1, n2, GCD_Num);
17    printf("time complexity :%d ",count);
18    return 0;
19 }

```

The console output shows the program execution with inputs 43 and 42, resulting in a GCD of 1 and a time complexity of 85. The program finished with exit code 0.

Q.9 Generate a program for Pascal triangle.

Estimate the time complexity for the row=5

```

          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1

```

Program

```

#include<stdio.h>

int main()
{
    int rows, coef = 1, space, i, j;
    int count=0;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    for (i = 0; i<rows; i++)
    {
        count++;
        for (space = 1; space <= rows - i; space++)
            printf(" ");
        count++;
        for (j = 0; j <= i; j++)
        {
            count++;
            if(j == 0 || i == 0){
                coef = 1;
                count++;
            }

            else
            {
                coef = coef * (i - j + 1) / j;
            }

```

```

        count++;

    printf("%4d", coef);

}

printf("\n");

count++;

}

printf("%d",count);

return 0;

}

```

The screenshot shows a web browser with the OnlineGDB online C compiler. The code in the editor is as follows:

```

1 #include<stdio.h>
2 int main()
3 {
4     int rows, coef = 1, space, i, j;
5     int count=0;
6     printf("Enter the number of rows: ");
7     scanf("%d", &rows);
8     while(count < rows)
9     {
10        printf("1 ");
11        for(space = 1; space < rows-count; space++)
12            printf(" ");
13        for(j = 1; j <= count; j++)
14            printf("%d ", coef);
15        coef = coef * j / count;
16        printf("\n");
17        count++;
18    }
19    printf("\n");
20    return 0;
21 }

```

The console output shows the program execution:

```

Enter the number of rows: 5
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
...Program finished with exit code 0
Press ENTER to exit console.

```

Q.10 Write a program to find the largest element value in an array.
Estimate the time complexity and no of comparison for the given set of values.

Program

```
#include <stdio.h>
```

```
int main() {
```

```
int n;
int count=0;
double arr[100];
printf("Enter the number of elements (1 to 100): ");
scanf("%d", &n);
count++;
for (int i = 0; i < n; ++i) {
    count++;
    printf("Enter number%d: ", i + 1);
    scanf("%lf", &arr[i]);
}
for (int i = 1; i < n; ++i) {
    count++;
    if (arr[0] < arr[i]) {
        arr[0] = arr[i];
    }
    count++;
}

printf("Largest element = %.2lf ", arr[0]);
printf("%d",count);
return 0;
}
```

```
1 #include <stdio.h>
2 int main() {
3     int n;
4     int count=0;
5     double arr[100];
6     //Enter the number of elements (1 to 100): 6
7     for (int i=0; i<n; i++) {
8         scanf("%lf", &arr[i]);
9     }
10    double max = arr[0];
11    for (int i=1; i<n; i++) {
12        if (arr[i] > max) {
13            max = arr[i];
14        }
15    }
16    printf("Largest element = %.2f\n", max);
17    return 0;
18 }
```

Enter the number of elements (1 to 100): 6
Enter number1: 45
Enter number2: 67
Enter number3: 78
Enter number4: 89
Enter number5: 54
Enter number6: 44
Largest element = 89.00 17
...Program finished with exit code 0
Press ENTER to exit console.

Q.11 Write a program to find the factorial (fact) of a number and to estimate time complexity.

Condition such as i. $n=0$, return 1 otherwise $\text{fact}(n-1) * n$

Program :

```
#include <stdio.h>
```

```
int main() {
```

```
    int n, i;
```

```
    int count=0;
```

```
    unsigned long long fact = 1;
```

```
    printf("Enter an integer: ");
```

```
    scanf("%d", &n);
```

```
    count++;
```

```
    if (n < 0)
```

```
        printf("Error! Factorial of a negative number doesn't exist.");
```

```
    else {
```

```
        for (i = 1; i <= n; ++i) {
```

```

        fact *= i;

        count++;

    }

    printf("Factorial of %d = %llu  ", n, fact);
    printf(" time compexity : %d ",count);

}

return 0;

}

```

The screenshot shows a web browser window with the OnlineGDB online compiler. The code in the editor is as follows:

```

1 #include <stdio.h>
2 int main() {
3     int n, i;
4     int count=0;
5     unsigned long long fact = 1;

```

The console output shows the program's execution:

```

Enter an integer: 12
Factorial of 12 = 479001600    time compexity : 13
...Program finished with exit code 0
Press ENTER to exit console.

```

Q.12 Write a program to print the first n perfect numbers. (Hint Perfect number means a positive integer that is equal to the sum of its proper divisors)

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0
2. N = 5
3. N = -2
4. N = -5

N = 0.2

PROGRAM

```
#include <stdio.h>
```

```
int main() {
```

```
    int n, i;
```

```
    int count=0;
```

```
    unsigned long long fact = 1;
```

```
    printf("Enter an integer: ");
```

```
    scanf("%d", &n);
```

```
    count++;
```

```
    if (n < 0)
```

```
        printf("Error! Factorial of a negative number doesn't exist.");
```

```
    else {
```

```
        for (i = 1; i <= n; ++i) {
```

```
            fact *= i;
```

```
            count++;
```

```
        }
```

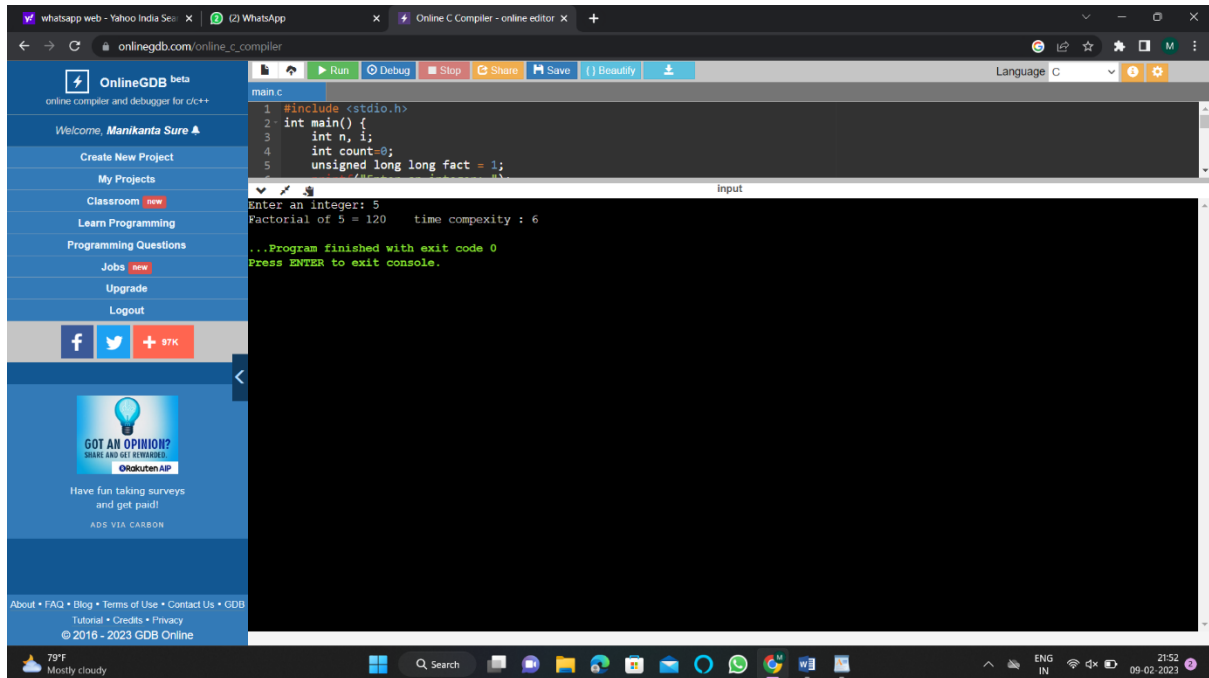
```
        printf("Factorial of %d = %llu  ", n, fact);
```

```
        printf(" time compexity : %d ",count);
```

```
}
```

```
return 0;
```

```
}
```



The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The page features a sidebar on the left with navigation links like 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Jobs', 'Upgrade', and 'Logout'. The main area displays a C program for calculating the factorial of a number. The code is as follows:

```
1 #include <stdio.h>
2 int main() {
3     int n, i;
4     int count=0;
5     unsigned long long fact = 1;
```

The console output shows the program execution: 'Enter an integer: 5', 'Factorial of 5 = 120', 'time complexity : 6', and '...Program finished with exit code 0'. The Windows taskbar at the bottom indicates the date and time as 09-02-2023, 21:52.

Q.13 Write a C program to check whether is a given input is a palindrome

PROGRAM

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int main() {
```

```
    char str[100];
```

```
    int i, length, flag = 0;
```

```
    printf("Enter a string: ");
```

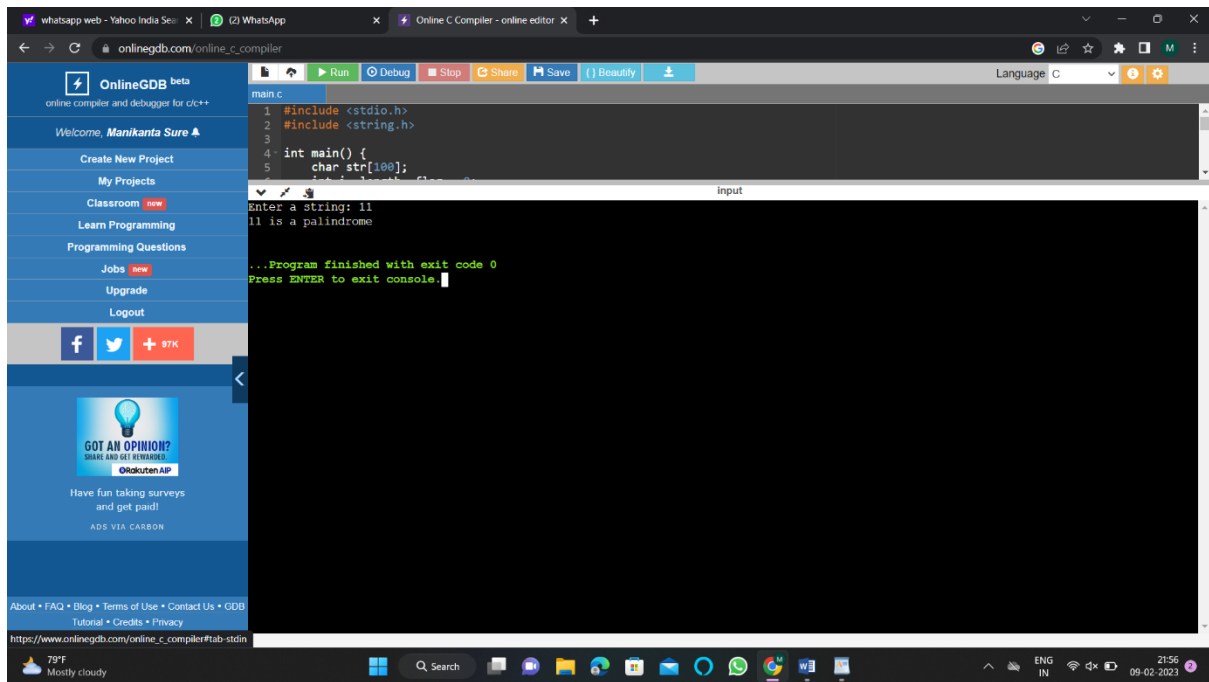
```
    scanf("%s", str);
```

```
length = strlen(str);

for(i=0; i < length ; i++){
    if(str[i] != str[length-i-1]){
        flag = 1;
        break;
    }
}

if (flag)
{
    printf("%s is not a palindrome and reverse\n", str);
}
else
{
    printf("%s is a palindrome\n", str);
}

return 0;
}
```



Q.14 Write a program to perform Bubble sort and estimate time Complexity

PROGRAM :

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int ele,count=0;
```

```
    printf("Enter tot element: ");
```

```
    scanf("%d",&ele);
```

```
    int arr[ele];
```

```
    printf("Enter the elements: ");
```

```
    for (int i = 0; i < ele; i++){
```

```
        count++;
```

```
scanf("%d",&arr[i]);  
}count++;
```

```
for (int i = 0; i < ele; i++)  
{  
count++;  
for (int j =i+1; j < ele; j++)  
{  
count++;  
if (arr[i]>arr[j])  
{  
count++;  
int temp=arr[i];  
count++;  
arr[i]=arr[j];  
count++;  
arr[j]=temp;  
count++;  
}  
}count++;
```

```
}count++;
```

```
printf("sorted array: ");  
for (int i = 0; i < ele; i++)  
{count++;  
count++;  
printf("%d ",arr[i]);
```

```

    }count++;

    printf("count: %d",count);

}

```

The screenshot shows the OnlineGDB website interface. The code editor contains the following C program:

```

main.c
12 // Enter the elements:
13 for (int i = 0; i < ele; i++){
14     count++;
15     scanf("%d",&arr[i]);
16 }count++;

```

The console output shows the program execution:

```

Enter tot element: 5
Enter the elements: 1
12
34
21
34
sorted array: 1 12 21 34 34 count: 42
...Program finished with exit code 0
Press ENTER to exit console.

```

Q.15 Write a program to print the reverse of a string. And estimate the time complexity

PROGRAM

```
#include<stdio.h>
```

```
int main(){
```

```
    char val[25];
```

```
    printf("enter the string: ");
```

```
    scanf("%s",&val);
```

```
    int count=0,c=0;
```

```
    while (val[count]!='\0'){
```

```
count++;
```

```
c++;
```

```
}c++;
```

```
for(int i=count-1;i>=0;i--){
```

```
    c++;
```

```
    printf("%c",val[i]);
```

```
}c++;
```

```
printf("\ncount: %d",c);
```

```
}
```

The screenshot displays the OnlineGDB web interface. The left sidebar contains navigation links such as 'Welcome, Manikanta Sure', 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Jobs', 'Upgrade', and 'Logout'. The main editor area shows a C program with the following code:

```
1 #include<stdio.h>
2
3 int main(){
4     char val[25];
5     scanf("%s", &val);
6     printf("%s\n", val);
7     int count = 0;
8     while(val[count] != '\0')
9         count++;
10    printf("Length of the string is: %d", count);
11}
```

The console output shows the program's execution:

```
main.c: In function 'main':
main.c:7:13: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[25]' [-Wformat=]
7 |     scanf("%s", &val);
  |             ^~
  |             |
  |             | char (*)[25]
  |             char *
enter the string: MAHESH BABU
MAHESH BABU
count: 14
...Program finished with exit code 0
Press ENTER to exit console.
```

The bottom status bar indicates the system temperature is 79°F, mostly cloudy, and the time is 22:02 on 09-02-2023.