

120CS0196

PROGRAMMING LABORATORY-7

CS1000

Assignment-6

Group-'P7' Section:'E'

Name: Prachi Nandi

Roll no: 120CS0196

Codes are uploaded on my GitHub account :
<https://github.com/prachi237/justC>

Assignment-7

1) Write a menu driven program in C, in which when the user enters a number it will ask the user to enter choice 1) 'p' for checking whether the number is prime or not, 2) 'f' for displaying all the prime factors of user entered number, 3) 'd' for displaying all the prime numbers, before user given number, 4) 'e' for exit from the program and all other choices should display wrong choice. The program should perform the operation according to the submitted choice.

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

```
int main()
```

```
{
```

```
char letter;
```

```
int a,i,factor=0,j,k;
```

```
printf("Enter p to check wether the no. is prime or not .\n");
```

```
printf("Enter f to display all the all the prime factors.\n");
```

```
printf("Enter d to display all the prime numbers\n");
```

```
printf("Enter e to exit the program \n");
```

```
printf("Enter your choice \n");
```

```
scanf("%c",&letter);
```

```
switch (letter)
```

```
{
```

```
case 'p':
```

```
printf("Enter the no. \n");
```

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

```
k=a/2;
```

```
for(i=2;i<=k;i++)
```

```
{
```

```
if(a%i == 0)
```

```
{
```

```
factor=1;
```

```

        break;
    }
}
if(a==1)
{
    printf("1 is neither prime nor composite.\n");
}
else
{
    if(factor==1)
    {
        printf("%d is not a prime no.\n",a);
        break;
    }
    else
    {
        printf("%d is a prime no.\n",a);
    }
}
break;
case 'f':
    printf("Enter the no. \n");
    scanf("%d",&a);
    k=a/2;
    for(i=2;i <= k;i++)
    {
        if(a%i == 0)
        {
            int fa=0;
            for(int o=2;o<i;o++)
            {
                if(i%o==0)
                {

```

```

        fa=1;
        break;
    }
}
if(fa==0)
{
    printf("%d ",i);
}
}
}
break;
case 'd':
    printf("Enter the no. \n");
    scanf("%d",&a);
    for(int j=1;j<a;j++)
    {
        int f=0;
        for(i=2;i <= j/2 ;i++)
        {
            if(j%i == 0)
            {
                f=1;
                break;
            }
        }
        if(f==0)
        {
            printf("%d ",j);
        }
    }
    break;
case 'e':
    printf("Exit\n");

```

```

        break;

    default :

        printf("enter right choice\n");

    }

}

return 0;

}

```

The screenshot displays the Dev-C++ IDE with a C++ program named 'q-1.cpp' open. The program is designed to collect student details (Name, Roll number, and Branch) and print them. The code includes standard headers, declares character arrays for name, roll number, and branch, and uses printf and scanf for input/output. The execution window shows the program running successfully, with the user inputting 'Prachi_Nandi' for the name, '120CS0196' for the roll number, and 'computer_Science' for the branch. The output matches the input, and the program exits with a return value of 0.

```

1 //WAP to enter the details of a student (Name, Roll_number, and Branch) as input through keyboard and print it.
2
3 #include<stdio.h>
4 int main()
5 {
6     char name[20];
7     char roll_no[10];
8     char branch[20];
9
10    printf("Enter your name please:");
11    scanf("%s",&name);
12    printf("\n Enter your Roll number:");
13    scanf("%s",&roll_no);
14    printf("\n Enter your branch:");
15    scanf("%s",&branch);
16
17    printf("\n Your name is %s",name);
18    printf("\n your roll number is %s",roll_no);
19    printf("\n your branch is %s",branch);
20
21    return 0;
22 }

```

Output Window (E:\just\Assignment6\q-1.exe):

```

Enter your name please:Prachi_Nandi
Enter your Roll number:120CS0196
Enter your branch:computer_Science

Your name is Prachi_Nandi
your roll number is 120CS0196
your branch is computer_Science

Process exited after 32.95 seconds with return value 0
Press any key to continue . . .

```

Compiler Output:

```

- Output Filename: E:\just\Assignment6\q-1.exe
- Output Size: 149.837890625 KiB
- Compilation Time: 0.20s

```

Status Bar: Line: 14 Col: 16 Sel: 0 Lines: 22 Length: 533 Insert Done parsing in 0.016 seconds

2) Write a program to insert new value into an sorted array. Clue: first find the position of the newly inserted element in the sorted array.

```
#include<stdio.h>

int main()
{
    int len;
    printf("Enter length of the array: ");
    scanf("%d",&len);
    int arr[len];
    for (int i = 0; i < len; i++)
    {
        printf("Enter the %dth value of array: ",i+1);
        scanf("%d",&arr[i]);
    }

    int k;
    printf("Enter the number which is to be inserted: ");
    scanf("%d",&k);
    int p;

    for (int i = 0; i < len; i++)
    {
        if(k<arr[i])
        {
            p = i;
            break;
        }
        else
        p = len;
    }
```

```
int arr2[len+1];
```

```
arr2[p]=k;
```

```
for (int i = 0; i < p; i++)
```

```
arr2[i] = arr[i];
```

```
for (int i = p+1; i < len+1; i++)
```

```
arr2[i] = arr[i-1];
```

```
printf("Array after inserting a new element \n");
```

```
for (int i = 0; i < len+1; i++)
```

```
printf("%d ",arr2[i]);
```

```
return 0;
```

```
}
```

The screenshot displays the Dev-C++ IDE with a C++ program open in the editor. The program is designed to find the prime factors of a user-input number. It includes a main function that prompts the user for a choice (p, f, d, e) and a number. Based on the choice, it either checks for primality, displays all prime factors, or displays all prime numbers up to the input number. The code uses a switch statement to handle these different cases. The output window shows the program's execution, including the prompts, user input (56), the list of prime factors (2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53), and the final message indicating the process exited after 11.41 seconds.

```
1 #include<stdio.h>
2
3 int main()
4 {
5     char letter;
6     int a,i,factor=0,j,k;
7     printf("Enter p to check wether the no. is prime or not .\n");
8     printf("Enter f to display all the all the prime factors.\n");
9     printf("Enter d to display all the prime numbers\n");
10    printf("Enter e to exit the program \n");
11    printf("Enter your choice \n");
12    scanf("%c",&letter);
13    switch (letter)
14    {
15        case 'p':
16            printf("Enter the no. \n");
17            scanf("%d",&a);
18            k=a/2;
19            for(i=2;i<=k;i++)
20            {
21                if(a%i == 0)
22                {
23                    factor=1;
24                    break;
25                }
26            }
27            if(a==1)
28            {
29                printf("1 is neither prime nor composite.\n");
30            }
31            else
32            {
33                if(factor==1)
34                {
35                    printf("%d is not a prime no.\n",a);
36                    break;
37                }
38            }
39        }
40    }
```

Output:

```
Enter p to check wether the no. is prime or not .
Enter f to display all the all the prime factors.
Enter d to display all the prime numbers
Enter e to exit the program
Enter your choice
d
Enter the no.
56
1 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53
.....
Process exited after 11.41 seconds with return value 0
Press any key to continue . . .
```

3) Write a C program to find the transpose of a matrix (Two-dimensional array).

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

```
int main()
```

```
{  
    int a[50][50], transpose[50][50], x, y, i=0, j=0;  
    printf("Enter rows and columns: ");  
    scanf("%i %i", &x, &y);  
  
    printf("\nEnter matrix elements:\n");  
    for (i = 0; i < x; ++i)  
    {  
        for (j = 0; j < y; ++j)  
        {  
            printf("Enter element a%i%i: ", i + 1, j + 1);  
            scanf("%d", &a[i][j]);  
        }  
    }  
}
```

```
printf("\nEntered matrix: \n");  
for (i = 0; i < x; ++i)  
{  
    for (j = 0; j < y; ++j)  
    {  
        printf("%d ", a[i][j]);  
        if (j == y - 1)  
            printf("\n");  
    }  
}
```

```
for (i = 0; i < x; ++i){  
    for (j = 0; j < y; ++j)  
    {  
        transpose[j][i] = a[i][j];  
    }  
}
```

```
printf("\nTranspose matrix: \n");  
for (i = 0; i < y; ++i)  
{  
    for (j = 0; j < x; ++j)  
    {  
        printf("%d ", transpose[i][j]);  
        if (j == x - 1)  
            printf("\n");  
    }  
}
```



```

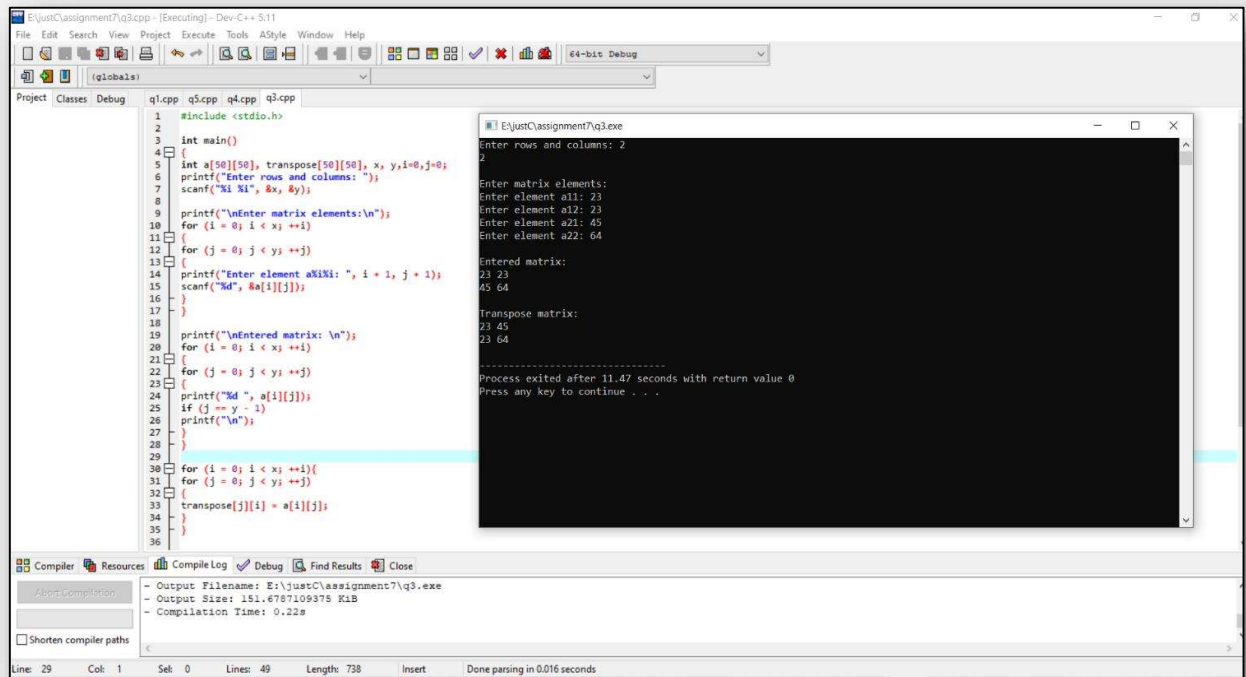
}
}

}

return 0;

}

```



4) Write a program to reverse a 4-digit number which is entered from keyboard. Input: 1234 , Output: 4321

```

#include<stdio.h>

int main()
{
    int num, reverse=0, rem;

    printf("Enter 4 digit number: ");

```

```
scanf("%i", &num);  
while(num!=0)  
{  
    rem=num%10;  
    reverse=reverse*10+rem;  
    num/=10;  
}  
printf("Reversed Number: %i",reverse);  
return 0;  
}
```

The screenshot displays the Dev-C++ IDE interface. The main window shows the source code for a C program that reverses a 4-digit number. The code is as follows:

```
1 #include<stdio.h>  
2 int main()  
3 {  
4     int num, reverse=0, rem;  
5     printf("Enter 4 digit number: ");  
6     scanf("%i", &num);  
7     while(num!=0)  
8     {  
9         rem=num%10;  
10        reverse=reverse*10+rem;  
11        num/=10;  
12    }  
13    printf("Reversed Number: %i",reverse);  
14    return 0;  
15 }  
16
```

The output window, titled "E:\justC\assignment7\q4.exe", shows the following output:

```
Enter 4 digit number: 1234  
Reversed Number: 4321  
-----  
Process exited after 11.76 seconds with return value 0  
Press any key to continue . . .
```

The status bar at the bottom indicates the current line is 12, column 2, with a selection of 0 lines and a length of 240 characters. The status bar also shows "Done parsing in 0 seconds".

5) What will be the output of the c program, and explain the output.

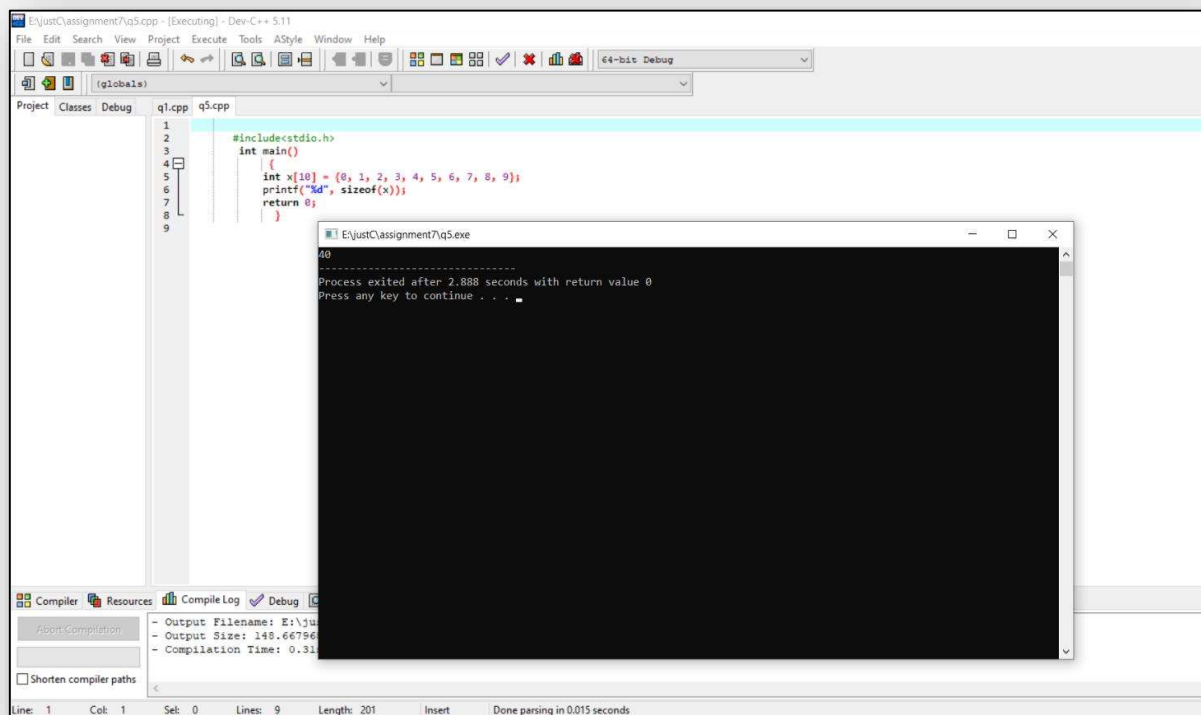
```
#include<stdio.h>

int main()
{
    int x[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

    printf("%d", sizeof(x));

    return 0;
}
```

OUTPUT: 40



6) Write a c program to display pyramid structure given below:

```
    1
  3  5
7  9 11
13 15 17 19
```

```
#include<stdio.h>
#include<string.h>
int main()
{
    int i,j,k,row, n=1;
    printf("enter row");
    scanf("%d",&row);
    printf("\n");
    for(i=1;i<=row;i++)
    {
        for(k=1;k<=row-i;k++)
        {
            printf(" ");
        }
        for(j=1;j<=i;j++)
        {
            printf("%d ",n);
            n=n+2;
        }
        printf("\n");
    }
    return 0;
}
```

The screenshot shows the Dev-C++ IDE with a C++ program named `q1.cpp` and its execution output in a separate window.

Source Code (q1.cpp):

```
1 #include<stdio.h>
2 int main()
3 {
4     int i,j,k,row,n=1;
5     printf("enter row");
6     scanf("%d",&row);
7     printf("\n");
8     for(i=1;i<=row;i++)
9     {
10         for(k=1;k<=row-i;k++)
11         {
12             printf(" ");
13         }
14         for(j=1;j<=i;j++)
15         {
16             printf("%d ",n);
17             n=n+2;
18         }
19         printf("\n");
20     }
21     return 0;
22 }
```

Execution Output:

```
enter row
6
1
3 5
7 9 11
13 15 17 19
21 23 25 27 29
31 33 35 37 39 41
-----
Process exited after 4.103 seconds with return value 0
Press any key to continue . . .
```

Compiler Output:

```
- Output Filename: E:\justC\assignment7\q1.exe
- Output Size: 150.009765625 KiB
- Compilation Time: 0.22s
```

Codes are uploaded on my GitHub account :

<https://github.com/prachi237/justC>

Submitted by: **Prachi Nandi** 120CS0196

THANK YOU