

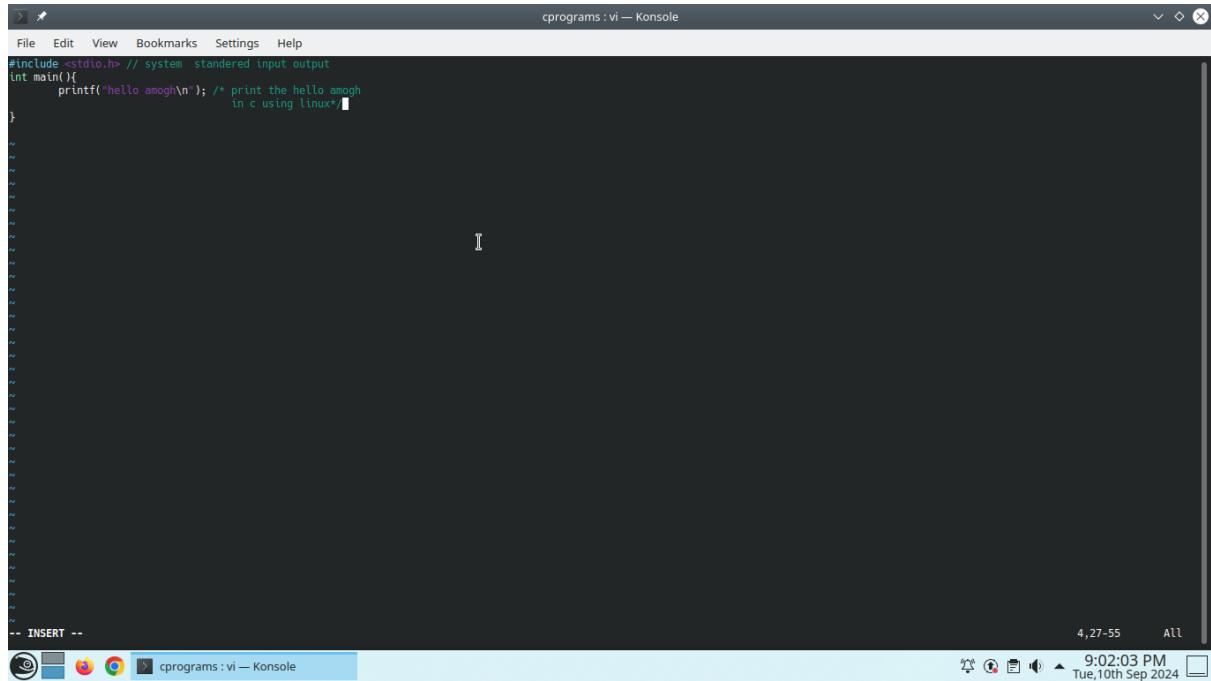
Name: Rushikesh Pundlik Sonekar

Roll No:-63

CPrograms

Q> Write a program to demonstrate different types of comments ?

- 1 Single line comment //
- 2 multi-line comment /*.....*/



The screenshot shows a terminal window titled "cprograms : vi — Konsole". The code editor displays the following C program:

```
#include <stdio.h> // system standered input output
int main(){
    printf("Hello amogh\n"); /* print the hello amogh
                                in c using linux*/
}
```

The terminal window includes a menu bar with File, Edit, View, Bookmarks, Settings, and Help. At the bottom, there are status indicators for INSERT mode, file name (cprograms : vi), and system time (9:02:03 PM, Tue, 10th Sep 2024).

Output :-

```
corporate@ACTS30:~/> ls
1.sh.save amogh.c amprog.c a.out coding.c cyberashvin.txt Downloads Linux Pictures rushikesh rushikesh.sh Videos
amoghaprof.c amogh.c.save and bin cprograms Desktop enoprog.c nano.save ruashikesh rushikesh1.txt rushikesh2.sh Templates
corporate@ACTS30:>>> cd cprograms
corporate@ACTS30:>/cprograms> vi day3.c
corporate@ACTS30:>/cprograms> gcc day3.c
corporate@ACTS30:>/cprograms> ./a.out
Hello amogh
corporate@ACTS30:>/cprograms> vi day3.c
corporate@ACTS30:>/cprograms> gcc day3.c
corporate@ACTS30:>/cprograms> ./a.out
Hello amogh
corporate@ACTS30:>/cprograms>
```

Q2> Write a program to demonstrate valid or invalid Identifier?

```
#include <stdio.h> // system standard input output
int main(){
    // valid identifiers
    int variable = 20;
    int var123 = 10;
    int _variable = 40;
    int var_name = 50;
    int maxValue = 30;

    // print valid identifier values
    printf("variable = %d\n", variable);
    printf("var123 = %d\n", var123);
    printf("_variable = %d\n", _variable);
    printf(" var_name = %d\n", var_name);
    printf("maxValue = %d\n", maxValue);
    return 0;
}
```

"day3.c" 19L, 436C

10,12 All

Output :-

```
Enter the number:10
corporate@ACTS30:~/cprograms> vi day3.c
corporate@ACTS30:~/cprograms> gcc day3.c
corporate@ACTS30:~/cprograms> ./a.out
Enter the number:10
you enterd : 5d /ncorporate@ACTS30:~/cprograms> gcc day3.c
corporate@ACTS30:~/cprograms> vi day3.c
corporate@ACTS30:~/cprograms> gcc day3.c

corporate@ACTS30:~/cprograms> gcc day3.c
day3.c: In function 'main':
day3.c:11:28: error: 'variable' undeclared (first use in this function); did you mean '_variable'?
    printf("variable = %d\n", _variable);
                           ^~~~~~
                           _variable
day3.c:11:28: note: each undeclared identifier is reported only once for each function it appears in
corporate@ACTS30:~/cprograms> vi day3.c
corporate@ACTS30:~/cprograms> gcc day3.c
corporate@ACTS30:~/cprograms> ./a.out
variable = 20
var123 = 10
_variable = 40
var_name = 50
maxvalue = 30
corporate@ACTS30:~/cprograms>
```

Invalid identifier:

```
[Running] cd "c:\C_programs\" && gcc Q2.c -o Q2 && "c:\C_programs\Q2
The value of _a is 10
The value of 0r is 6
The value of   is $rohit 10

[Done] exited with code=0 in 0.376 seconds
```

Q3> demonstrate 5 key words ?

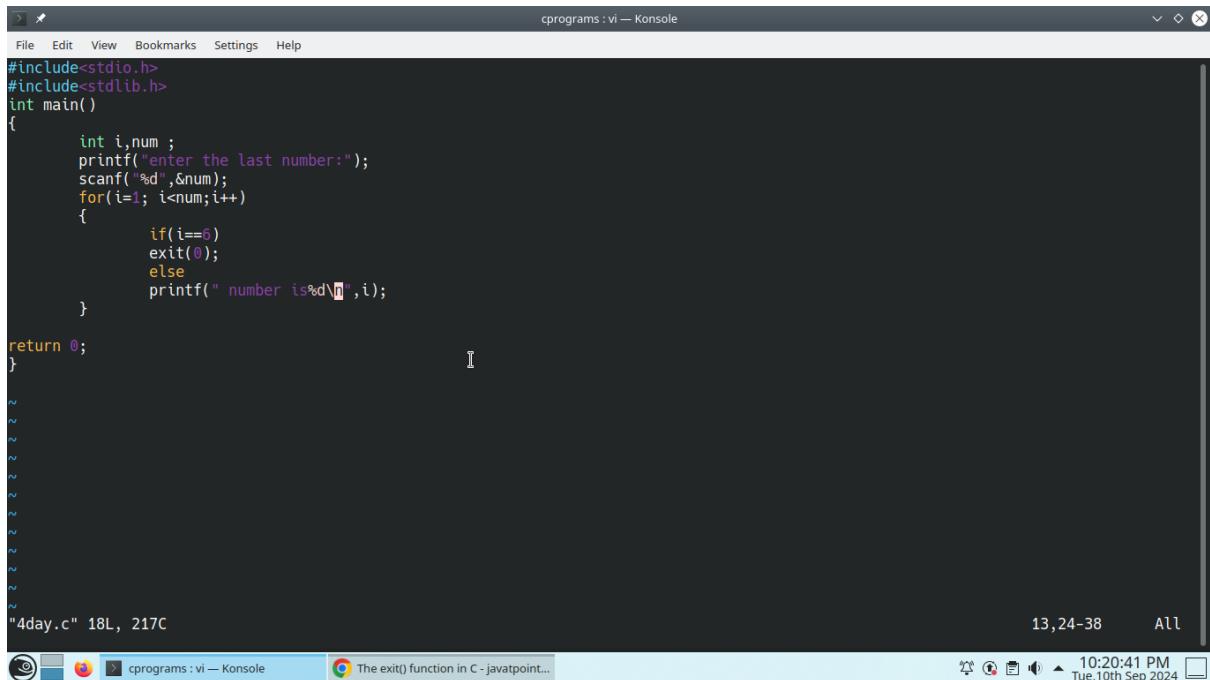
A)Break

```
cprograms : vi — Konsole
File Edit View Bookmarks Settings Help
#include<stdio.h>
int main()
{
    printf("breaking for loop \n");
    for(int i=1;i<5; i++)
    {
        if(i==3)
        {
            break;
        }
        else{
            printf("%d\n",i);
        }
    }
    return 0;
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
day31.c" 16L, 168C
12,15-36 All
10:00:13 PM Tue,10th Sep 2024
```

Output :-

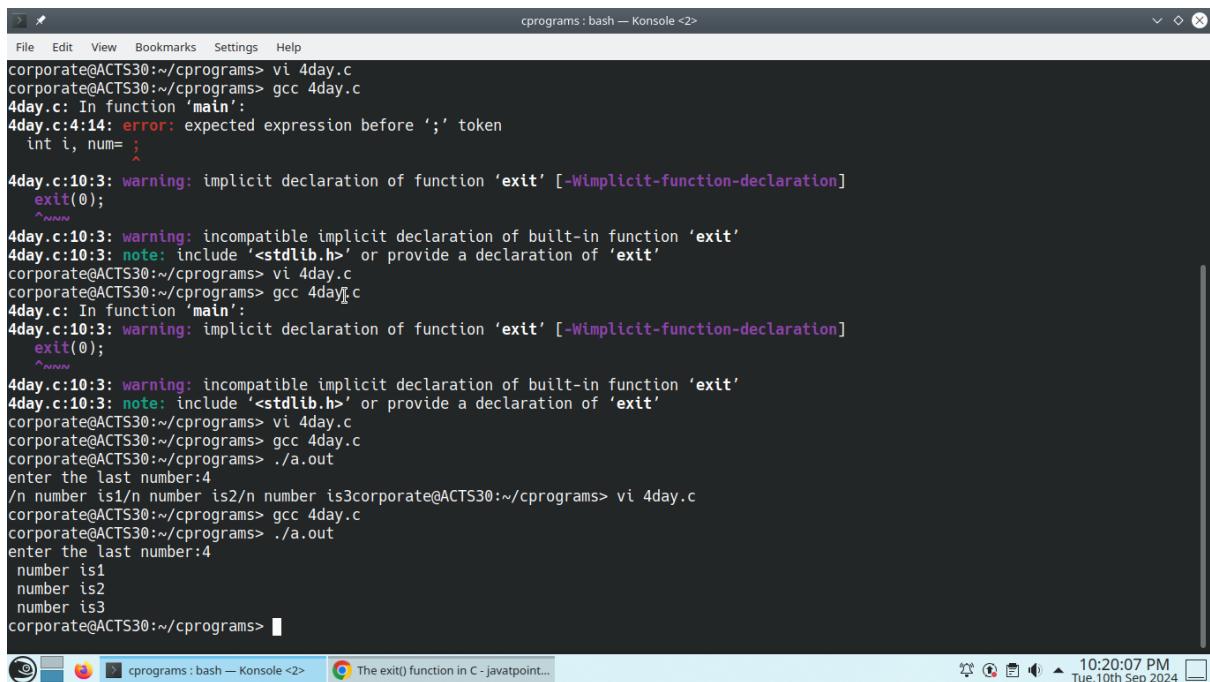
```
cprograms : bash — Konsole
File Edit View Bookmarks Settings Help
12corporate@ACTS30:~/cprograms> cat day31.c
#include<stdio.h>
int main()
{
    printf("breaking for loop \n");
    for(int i=1;i<5; i++)
    {
        if(i==3)
        {
            break;
        }
        else{
            printf("%d",i);
        }
    }
    return 0;
}
corporate@ACTS30:~/cprograms> gcc day31.c
corporate@ACTS30:~/cprograms> ./a.out3
bash: ./a.out3: No such file or directory
corporate@ACTS30:~/cprograms> ./a.out
breaking for loop
1
2
corporate@ACTS30:~/cprograms>
```

B> exit



```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int i,num ;
    printf("enter the last number:");
    scanf("%d",&num);
    for(i=1; i<num;i++)
    {
        if(i==5)
            exit(0);
        else
            printf(" number is%d\n",i);
    }
    return 0;
}
~
```

Output :-



```
corporate@ACTS30:~/cprograms> vi 4day.c
corporate@ACTS30:~/cprograms> gcc 4day.c
4day.c: In function 'main':
4day.c:4:14: error: expected expression before ';' token
  int i, num= ;
^~~~~
4day.c:10:3: warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]
  exit(0);
^~~~~
4day.c:10:3: warning: incompatible implicit declaration of built-in function 'exit'
4day.c:10:3: note: include '<stdlib.h>' or provide a declaration of 'exit'
corporate@ACTS30:~/cprograms> vi 4day.c
corporate@ACTS30:~/cprograms> gcc 4day.c
4day.c: In function 'main':
4day.c:10:3: warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]
  exit(0);
^~~~~
4day.c:10:3: warning: incompatible implicit declaration of built-in function 'exit'
4day.c:10:3: note: include '<stdlib.h>' or provide a declaration of 'exit'
corporate@ACTS30:~/cprograms> vi 4day.c
corporate@ACTS30:~/cprograms> gcc 4day.c
corporate@ACTS30:~/cprograms> ./a.out
enter the last number:4
1 number is1/n number is2/n number is3/corporate@ACTS30:~/cprograms> vi 4day.c
corporate@ACTS30:~/cprograms> gcc 4day.c
corporate@ACTS30:~/cprograms> ./a.out
enter the last number:4
number is1
number is2
number is3
```

C> Extern

The screenshot shows a terminal window titled "rushikesh@rushikesh-VirtualBox: ~". The window title bar also displays "GNU nano 7.2" and "extern.c". The terminal background is dark purple. The code in the editor is:

```
GNU nano 7.2
#include<stdio.h>
extern int global_variable;
int main(){
    global_variable =10;
    printf("global_variable:%d\n",global_variable);
    return 0;
}
int global_variable;
```

At the bottom of the terminal window, there is a menu bar with various keyboard shortcuts for file operations like Help, Exit, Write Out, Read File, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, and Set Mark.

Output:-

The screenshot shows a terminal window titled "rushikesh@rushikesh-VirtualBox: ~". The terminal background is dark purple. The command history and output are:

```
rushikesh@rushikesh-VirtualBox:~$ nano extern.c
rushikesh@rushikesh-VirtualBox:~$ gcc extern.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
global_variable:10
rushikesh@rushikesh-VirtualBox:~$
```

D> Continue

The screenshot shows a terminal window titled "GNU nano 7.2" with the command "continue.c" open. The file contains the following C code:

```
#include<stdio.h>
int main(){
    for(int i=0;i<10;i++){
        if(i==5){
            continue;
        }
        printf("%d\n",i);
    }
    return 0;
}
```

The terminal window has a dark background with light-colored text. It includes a standard set of keyboard shortcuts at the bottom:

- Help (^G)
- Exit (^X)
- Write Out (^O)
- Where Is (^W)
- Cut (^K)
- Paste (^U)
- Execute (^T)
- Justify (^J)
- Location (^C)
- Go To Line (^L)
- Undo (M-U)
- Redo (M-E)
- Set Mark (M-A)
- Copy (M-G)

Output:

The screenshot shows a terminal window with the following session:

```
rushikesh@rushikesh-VirtualBox:~$ nano continue.c
rushikesh@rushikesh-VirtualBox:~$ gcc continue.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
0
1
2
3
4
5
6
7
8
9
rushikesh@rushikesh-VirtualBox:~$ nano continue.c
rushikesh@rushikesh-VirtualBox:~$ gcc continue.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
```

The terminal window has a dark background with light-colored text. It includes a standard set of keyboard shortcuts at the bottom:

- Help (^G)
- Exit (^X)
- Write Out (^O)
- Where Is (^W)
- Cut (^K)
- Paste (^U)
- Execute (^T)
- Justify (^J)
- Location (^C)
- Go To Line (^L)
- Undo (M-U)
- Redo (M-E)
- Set Mark (M-A)
- Copy (M-G)

E> Return

A screenshot of a Linux desktop environment. On the left is a dock with icons for a browser, file manager, terminal, help, dash, and system settings. The main window is a terminal with a dark background and white text. The terminal title bar says "rushikesh@rushikesh-VirtualBox: ~". The date and time "Sep 16 11:35" are at the top right. The terminal content shows a C program named "return1.c" with 9 lines and 90 bytes:

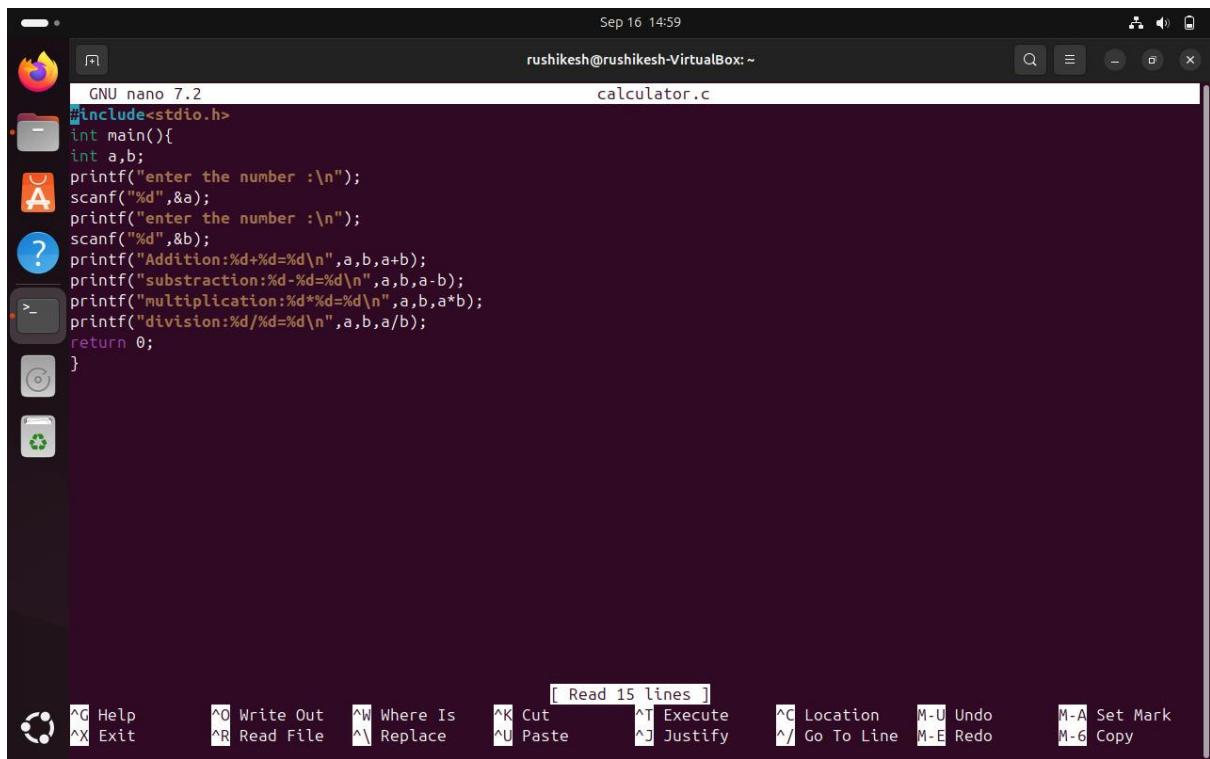
```
#include<stdio.h>
int main(){
    char grade ='a';
    printf("grade:%c\n",grade);
    return 0;
}
```

Output:-

A screenshot of a Linux desktop environment, identical to the one above, showing the same terminal window. The terminal title bar says "rushikesh@rushikesh-VirtualBox: ~". The date and time "Sep 16 11:36" are at the top right. The terminal content shows the execution of the C program:

```
rushikesh@rushikesh-VirtualBox:~$ vi return1.c
rushikesh@rushikesh-VirtualBox:~$ gcc return1.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
grade:a
rushikesh@rushikesh-VirtualBox:~$ vi return1.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
```

Q4. Write a C program to perform basic calculation.

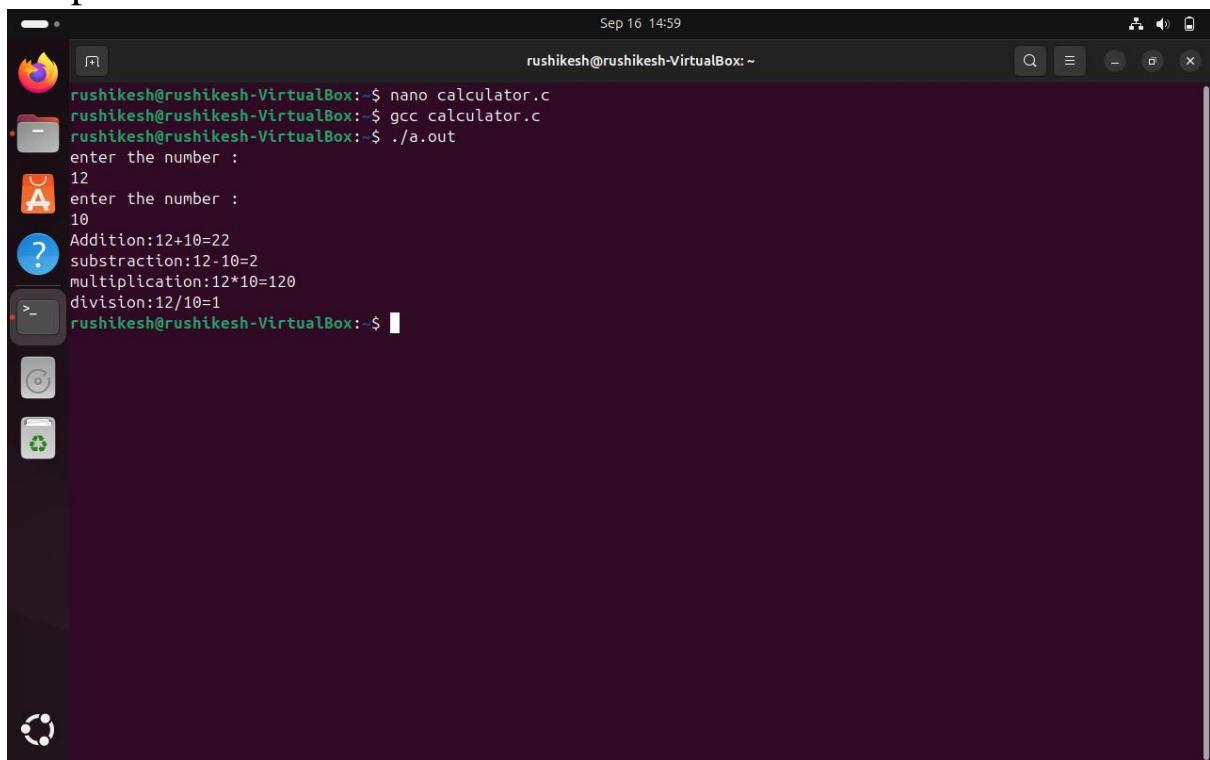


```
GNU nano 7.2                                     rushikesh@rushikesh-VirtualBox: ~
#include<stdio.h>
int main(){
int a,b;
printf("enter the number :\n");
scanf("%d",&a);
printf("enter the number :\n");
scanf("%d",&b);
printf("Addition:%d+%d=%d\n",a,b,a+b);
printf("subtraction:%d-%d=%d\n",a,b,a-b);
printf("multiplication:%d*%d=%d\n",a,b,a*b);
printf("division:%d/%d=%d\n",a,b,a/b);
return 0;
}
```

[Read 15 lines]

^G Help ^X Exit ^O Write Out ^R Read File ^W Where Is ^K Cut ^U Paste ^T Execute ^C Location ^I Go To Line M-U Undo M-E Redo M-A Set Mark M-6 Copy

Output:-



```
rushikesh@rushikesh-VirtualBox:~$ nano calculator.c
rushikesh@rushikesh-VirtualBox:~$ gcc calculator.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter the number :
12
enter the number :
10
Addition:12+10=22
subtraction:12-10=2
multiplication:12*10=120
division:12/10=1
rushikesh@rushikesh-VirtualBox:~$
```

Q5. Write C program to find length of data type .

The screenshot shows a terminal window titled "identifier.c" with the following content:

```
GNU nano 7.2
#include<stdio.h>
int main(){
    printf("size of Intger :%ld bytes. \n",sizeof(int));
    printf("size of float :%ld bytes. \n",sizeof(float));
    printf("size of double :%ld bytes. \n",sizeof(double));
    printf("size of b00l :%ld bytes. \n",sizeof(_Bool));
    return 0;
}
```

The terminal window has a dark theme and includes a menu bar at the top with options like Help, Exit, Write Out, Read File, Replace, Cut, Paste, Execute, Justify, Location, Go To Line, Undo, Redo, Set Mark, and Copy. The status bar at the bottom shows "Sep 16 15:22" and the user "rushikesh@rushikesh-VirtualBox:~".

Output:-

The screenshot shows a terminal window with the following output:

```
int         long unsigned int
           %ld
identifier.c:4:25: warning: format '%d' expects argument of type 'int', but argument 2 has type 'long unsigned int' [-Wformat]
  4 | printf("size of float :%d bytes. \n",sizeof(float));
           ^~~~~~
           |         |
           int         long unsigned int
           %ld
identifier.c:5:26: warning: format '%d' expects argument of type 'int', but argument 2 has type 'long unsigned int' [-Wformat]
  5 | printf("size of double :%d bytes. \n",sizeof(double));
           ^~~~~~
           |         |
           int         long unsigned int
           %ld
identifier.c:6:24: warning: format '%d' expects argument of type 'int', but argument 2 has type 'long unsigned int' [-Wformat]
  6 | printf("size of b00l :%d bytes. \n",sizeof(_Bool));
           ^~~~~~
           |         |
           int         long unsigned int
           %ld
rushikesh@rushikesh-VirtualBox:~$ nano identifier.c
rushikesh@rushikesh-VirtualBox:~$ gcc identifier.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
size of Intger :4 bytes.
size of float :4 bytes.
size of double :8 bytes.
size of b00l :1 bytes.
rushikesh@rushikesh-VirtualBox:~$
```

Q6. Write a C program to check return type of printf() method.

The screenshot shows a terminal window titled "GNU nano 7.2" with the file "printf.c" open. The code prints the size of the return type of printf:

```
#include<stdio.h>
int main(){
// print the size of return type of printf
    printf(" the return value of print is :int\n");
    printf(" the return type value of printf is: %ld bytes\n", sizeof(int));
    return 0;
}
```

The terminal window has a dark theme with icons on the left side. The bottom status bar shows keyboard shortcuts for various functions.

Output:

The screenshot shows a terminal window with the following session:

```
rushikesh@rushikesh-VirtualBox:~$ nano printf.c
rushikesh@rushikesh-VirtualBox:~$ gcc printf.c
printf.c: In function 'main':
printf.c:6:48: warning: format '%d' expects argument of type 'int', but argument 2 has type 'long unsigned int' [-Wformat
t=]
  6 |     printf(" the return type value of printf is: %d bytes\n", sizeof(int));
   |          ~~~~~~
   |          |
   |          int      long unsigned int
   |          %ld
%ld

rushikesh@rushikesh-VirtualBox:~$ ./a.out
the return value of print is :int
the return type value of printf is: 4 bytes
rushikesh@rushikesh-VirtualBox:~$
```

The terminal window has a dark theme with icons on the left side. The bottom status bar shows keyboard shortcuts for various functions.

-----Day2-----

Q1) write C program to Demonstrate following operator:

- a)! (Not) operator.
- b)<<,>>(shift) operator.
- c)== (Comparison) Operator.
- d) ++,-- (Unary) Operator.
- e)||,&&(logical operator).

The screenshot shows a terminal window titled "cprogram001 : vi — Konsole". The code in the terminal is as follows:

```
//Q1 Write a program to demonstrate following operrator in c
/*
 *!operator
 *shift Operator
 *Comparision operator
 *Unary operator
 *logical operatore

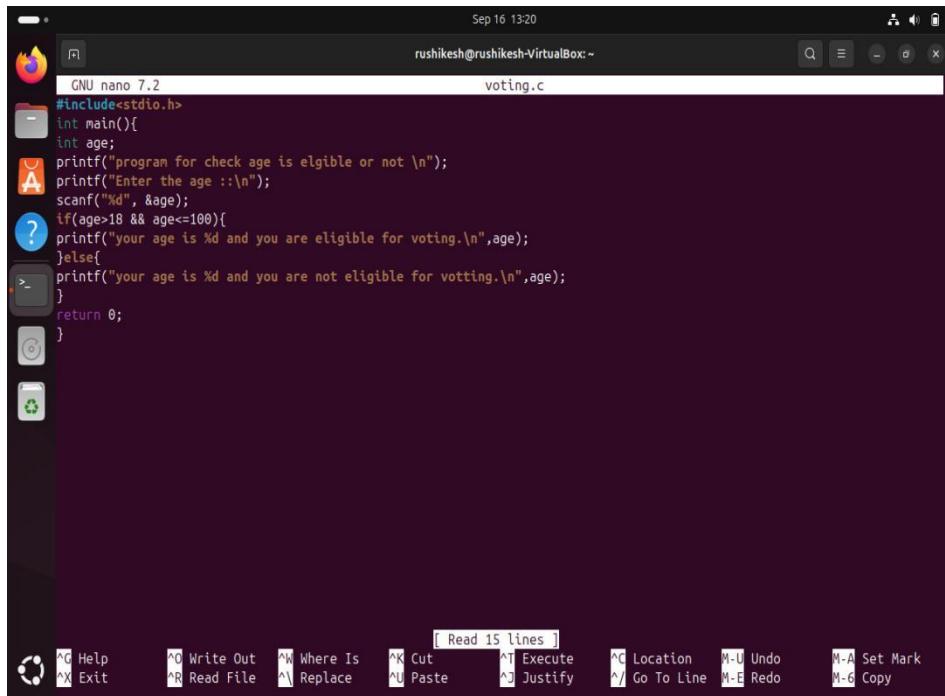
 */
#include <stdio.h>

int main(){
    int b=6;
    //Using !operator
    printf("! operator:%d\n",!b);
    //Shift(<<) operator
    printf("<< operator:%d\n",b<<5);
    //Shift(>>) operator
    printf(">> operator:%d\n",b>>3);
    //Comparison Operotor(==) operator
    printf("Comparison Operator:%d\n",b==6);
    //Unary Operator(++) operator
    printf("unary operator:%d\n",++b);
    //Unary Operator(-- operator
    printf("Unary operator:%d\n",--b);
    //Logical Operaor(&&) operator
    printf("Logical and Operator:%d\n",b==6 && b>3);
    //Logical(||) operator
    printf("Logical or operator:%d\n",b==6 || b>9);
}
```

Output:-

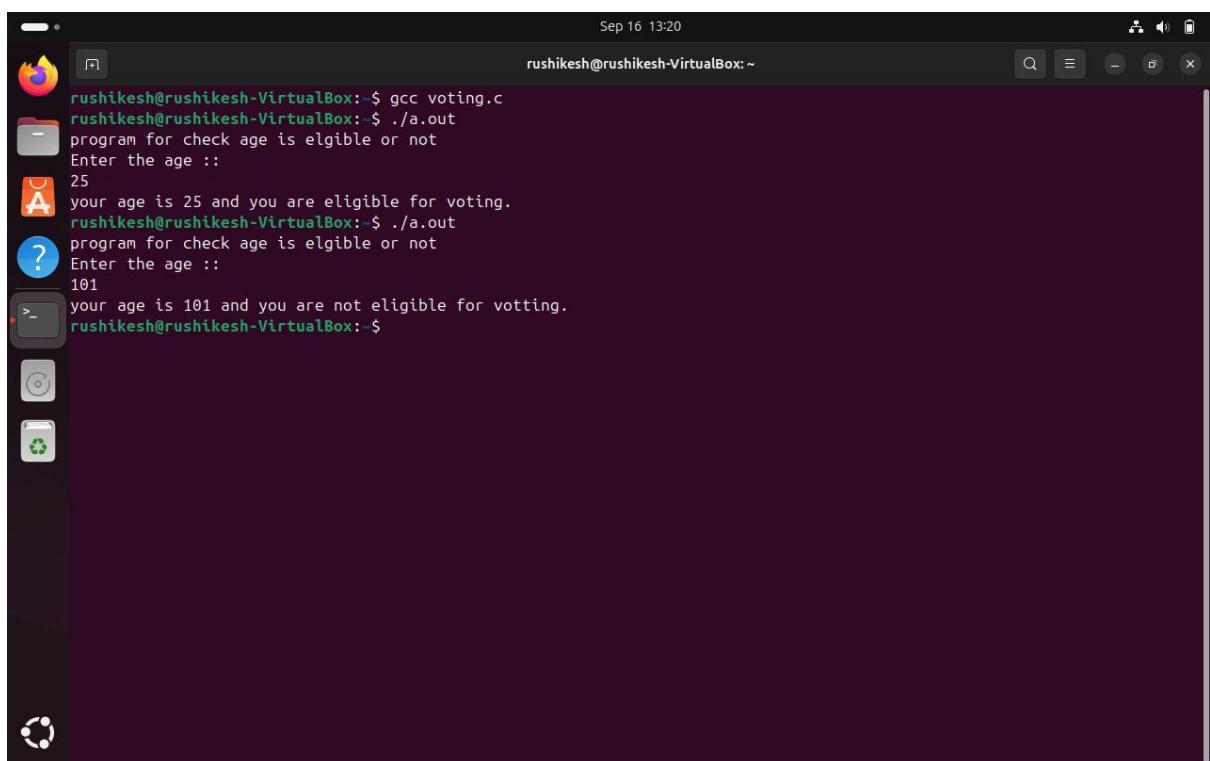
```
corporate@ACTS31:~/cprogram001> vi Q1.c
corporate@ACTS31:~/cprogram001> gcc Q1.c
corporate@ACTS31:~/cprogram001> ./a.out
! operator:0
<< operator:192
>> operator:0
Comparison Operator:1
unary operator:7
Unary operator:6
Logical and Operator:1
Logical or operator:1
corporate@ACTS31:~/cprogram001>
```

Q2. Write a c program to check a person is eligible for voting or not.



```
GNU nano 7.2
#include<stdio.h>
int main(){
int age;
printf("program for check age is eligible or not \n");
printf("Enter the age ::\n");
scanf("%d", &age);
if(age>18 && age<=100){
printf("your age is %d and you are eligible for voting.\n",age);
}else{
printf("your age is %d and you are not eligible for voting.\n",age);
}
return 0;
}
```

Output:-



```
Sep 16 13:20
rushikesh@rushikesh-VirtualBox:~$ gcc voting.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
program for check age is eligible or not
Enter the age :::
25
your age is 25 and you are eligible for voting.
rushikesh@rushikesh-VirtualBox:~$ ./a.out
program for check age is eligible or not
Enter the age :::
101
your age is 101 and you are not eligible for voting.
rushikesh@rushikesh-VirtualBox:~$
```

Q3. Write a c program to check the speed is above 60 or not.

The screenshot shows a Linux desktop environment. In the background, there is a terminal window titled "rushikesh@rushikesh-VirtualBox: ~". In the foreground, there is a nano editor window titled "speed.c". The code in the nano window is:

```
GNU nano 7.2
#include<stdio.h>
int main (){
int speed;
printf("\n check permission to drive limit\n");
printf("\n enter the speed of your bike:\n");
scanf("%d",&speed);
if(speed>=0 && speed<=60){
printf("your speed is %d and you have permission to drive \n",speed);
}else{
printf("your speed is %d and you have not permission to drive \n",60);
}
return 0;
}
```

The nano editor interface includes standard keyboard shortcuts at the bottom:

- ^G Help
- ^X Exit
- ^O Write Out
- ^R Read File
- ^W Where Is
- ^A Replace
- ^K Cut
- ^U Paste
- ^T Execute
- ^J Justify
- ^C Location
- ^L Go To Line
- M-U Undo
- M-E Redo
- M-A Set Mark
- M-C Copy

Output:-

The screenshot shows a terminal window with the following session:

```
sep 16 13:32
rushikesh@rushikesh-VirtualBox: ~$ nano speed.c
rushikesh@rushikesh-VirtualBox: ~$ gcc speed.c
speed.c: In function 'main':
speed.c:10:1: error: 'esle' undeclared (first use in this function)
  10 | esle{
      | ^
      |
      ;
rushikesh@rushikesh-VirtualBox: ~$ ./a.out
check permission to drive limit
enter the speed of your bike:
56
your speed is 56 and you have permission to drive
rushikesh@rushikesh-VirtualBox: ~$ ./a.out
check permission to drive limit
enter the speed of your bike:
80
your speed is 80 and you have not permission to drive
rushikesh@rushikesh-VirtualBox: ~$
```

4) Write a program to check the given is prime or not.

```
cprogram001 : vi — Konsole
File Edit View Bookmarks Settings Help
//Q4.Write a program for to check the given number is prime or not
#include <stdio.h>
int main()
{
    int i, num, temp = 0;
    printf("Enter any number to Check for Prime: ");
    scanf("%d", &num);
    for (i = 2; i <= num / 2; i++)
    {
        if (num % i == 0)
        {
            temp++;
            break;
        }
    }
    if (temp == 0 && num != 1)
    {
        printf("%d is a Prime number\n", num);
    }
    else
    {
        printf("%d is not a Prime number\n", num);
    }
}
return 0;
~
```

Output:-

```
cprogram001 : bash — Konsole
File Edit View Bookmarks Settings Help
corporate@ACTS31:~/cprogram001> vi Q4.c
corporate@ACTS31:~/cprogram001> gcc Q4.c
corporate@ACTS31:~/cprogram001> ./a.out
Enter any number to Check for Prime: 13
13 is a Prime number
corporate@ACTS31:~/cprogram001> ./a.out
Enter any number to Check for Prime: 8
8 is not a Prime number
corporate@ACTS31:~/cprogram001>
```

Q5> WAP to check applicant is eligible for a) trainee b) full stack
c)database admin d) others.

The screenshot shows a terminal window titled "GNU nano 7.2 applicant.c". The code is a C program that prompts the user for a skill level (1, 2, 3, or 4) and prints whether they are eligible for a trainee, full stack developer, database administrator, or others. The terminal interface includes standard nano key bindings at the bottom.

```
GNU nano 7.2 applicant.c
#include<stdio.h>
#include<string.h>
int main(){
int skill=0;
printf("choose your skill \n");
printf("1) C,C++ \n 2)full stack \n 3)database adminstrator \n 4) others\n");
printf("enter the number of which you have skills");
scanf("%d",&skill);
if (skill==1)
{
printf("you are eligible for :trainee\n");
}
if(skill==2)
{
printf("you are eligible for :full stack\n");
}
if(skill==3)
{
printf("you are eligible for :database admin\n");
}
if(skill==4)
{
printf("you are not eligible  :\n");
}
return 0;
}
```

[Read 28 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^I Go To Line M-E Redo M-A Set Mark M-6 Copy

Output:

The screenshot shows a terminal window with the following session:

```
rushikesh@rushikesh-VirtualBox:~$ nano applicant.c
rushikesh@rushikesh-VirtualBox:~$ gcc applicant.c
applicant.c: In function 'main':
applicant.c:8:13: error: 'skills' undeclared (first use in this function); did you mean 'skill'?
  8 |     scanf("%d",&skills);
      |           ^
      |           skill
applicant.c:8:13: note: each undeclared identifier is reported only once for each function it appears in
rushikesh@rushikesh-VirtualBox:~$ nano applicant.c
rushikesh@rushikesh-VirtualBox:~$ gcc applicant.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
choose your skill
1) C,C++
2)full stack
3)database adminstrator
4) others
enter the number of which you have skills2
you are eligible for :full stack
rushikesh@rushikesh-VirtualBox:~$
```

Q6)Write a program to print odd number between 0 to 10.

cprogram001 : vi — Konsole

File Edit View Bookmarks Settings Help

```
//Q7.Write a program to print odd number between to 1 to 10:  
#include <stdio.h>  
int main(){  
    printf("Print odd numbers between 1 to 10:\n");  
    for(int i=1; i<=10;i++){  
        if(i%2==1){  
            printf("%d\n",i);  
        }  
    }  
  
    return 0;  
}
```

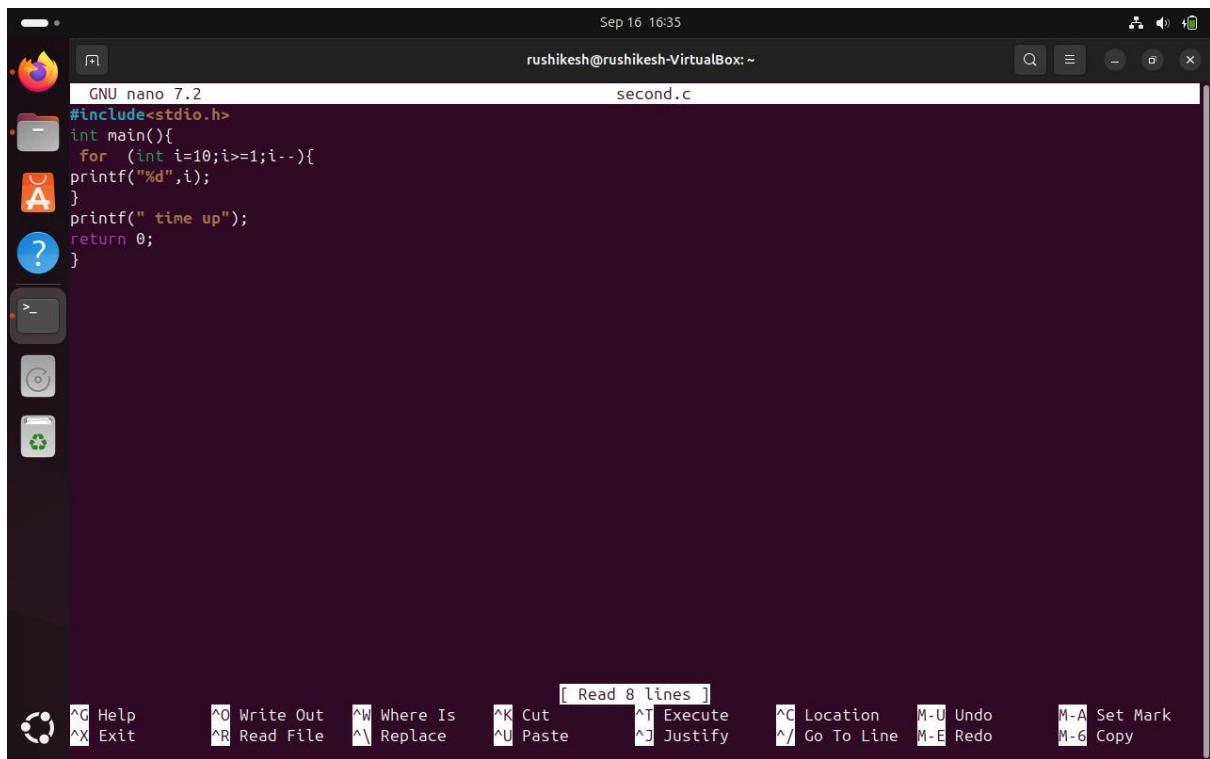
Output:-

cprogram001 : bash — Konsole

File Edit View Bookmarks Settings Help

```
corporate@ACTS31:~/cprogram001> cd cprogram001  
corporate@ACTS31:~/cprogram001> vi Q7.c  
corporate@ACTS31:~/cprogram001> gcc Q7.c  
Q7.c: In function ‘main’:  
Q7.c:5:18: error: expected expression before '=' token  
    for(int i=1; i<=10;i++){  
                  ^  
corporate@ACTS31:~/cprogram001> vi Q7.c  
corporate@ACTS31:~/cprogram001> gcc Q7.c  
corporate@ACTS31:~/cprogram001> ./a.out  
Print odd numbers between 1 to 10:  
1  
3  
5  
7  
9  
corporate@ACTS31:~/cprogram001> vi Q7.c  
corporate@ACTS31:~/cprogram001> gcc Q7.c  
corporate@ACTS31:~/cprogram001> ./a.out  
Print odd numbers between 1 to 10:  
1  
3  
5  
7  
9  
corporate@ACTS31:~/cprogram001>
```

Q7) WAP to display last 10 seconds and the message is time up.

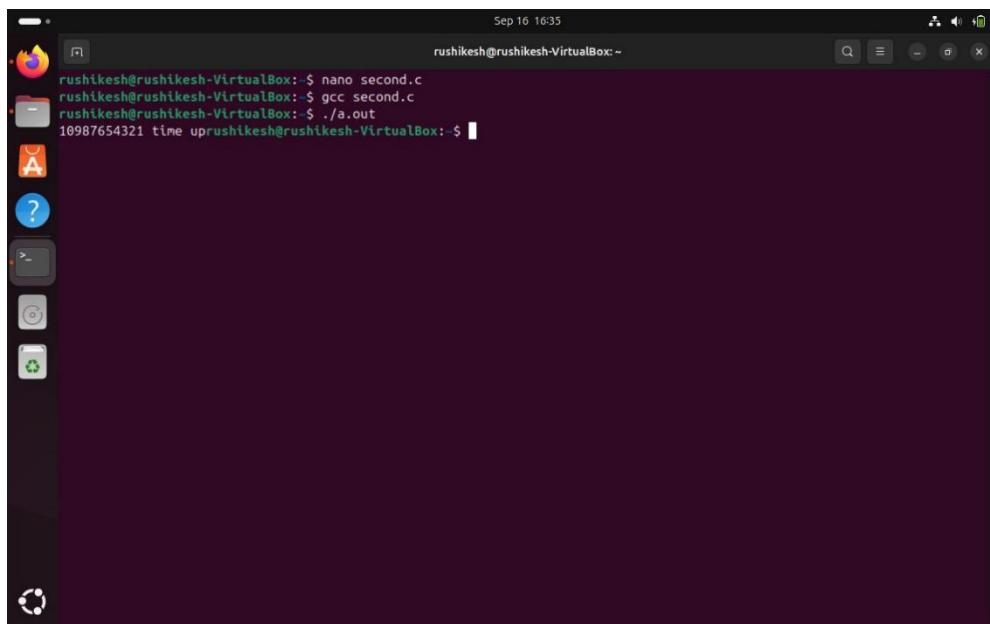


```
GNU nano 7.2
#include<stdio.h>
int main(){
    for (int i=10;i>=1;i--){
        printf("%d",i);
    }
    printf(" time up");
    return 0;
}
```

[Read 8 lines]

^G Help ^X Exit ^O Write Out ^R Read File ^W Where Is ^K Cut ^U Paste ^T Execute ^J Justify ^C Location ^I Go To Line M-U Undo M-E Redo M-A Set Mark M-6 Copy

Output:-



```
rushikesh@rushikesh-VirtualBox: $ nano second.c
rushikesh@rushikesh-VirtualBox: $ gcc second.c
rushikesh@rushikesh-VirtualBox: $ ./a.out
10987654321 time uprushikesh@rushikesh-VirtualBox: $
```

Q8) WAP use break for loop

Output:-

```
cprograms : bash — Konsole
File Edit View Bookmarks Settings Help
12corporate@ACTS30:~/cprograms> cat day31.c
#include<stdio.h>
int main()
{
    printf("breaking for loop \n");
    for(int i=1;i<5; i++)
    {
        if(i==3)
        {
            break;
        }
        else{
            printf("%d",i);
        }
    }
    return 0;
}
corporate@ACTS30:~/cprograms> gcc day31.c
corporate@ACTS30:~/cprograms> ./a.out3
bash: ./a.out3: No such file or directory
corporate@ACTS30:~/cprograms> ./a.out
breaking for loop
12corporate@ACTS30:~/cprograms> vim day31.c
corporate@ACTS30:~/cprograms> gcc day31.c
corporate@ACTS30:~/cprograms> ./a.out
breaking for loop
1
2
corporate@ACTS30:~/cprograms>
```

Q9) Write a program to print the day of the week using Switch case.

The screenshot shows a terminal window titled "week.c" running on a Linux system. The window has a dark background with light-colored text. The terminal title bar includes the date "Sep 16 17:15" and the user "rushikesh@rushikesh-VirtualBox: ~". The main area of the terminal displays the following C code:

```
GNU nano 7.2
int main(){
int n;
printf("enter the number the day:");
scanf("%d",&n);
switch (n){
case 1:
printf("monday\n");
break;
case 2:
printf("tuesday\n");
break;
case 3:
printf("wednesday\n");
break;
case 4:
printf("thursday\n");
break;
case 5:
printf("friday\n");
break;
case 6:
printf("saturday\n");
break;
case 7:
printf("sunday\n");
default:
printf(" invalid input\n");
}
}
```

The terminal also shows a set of keyboard shortcuts at the bottom:

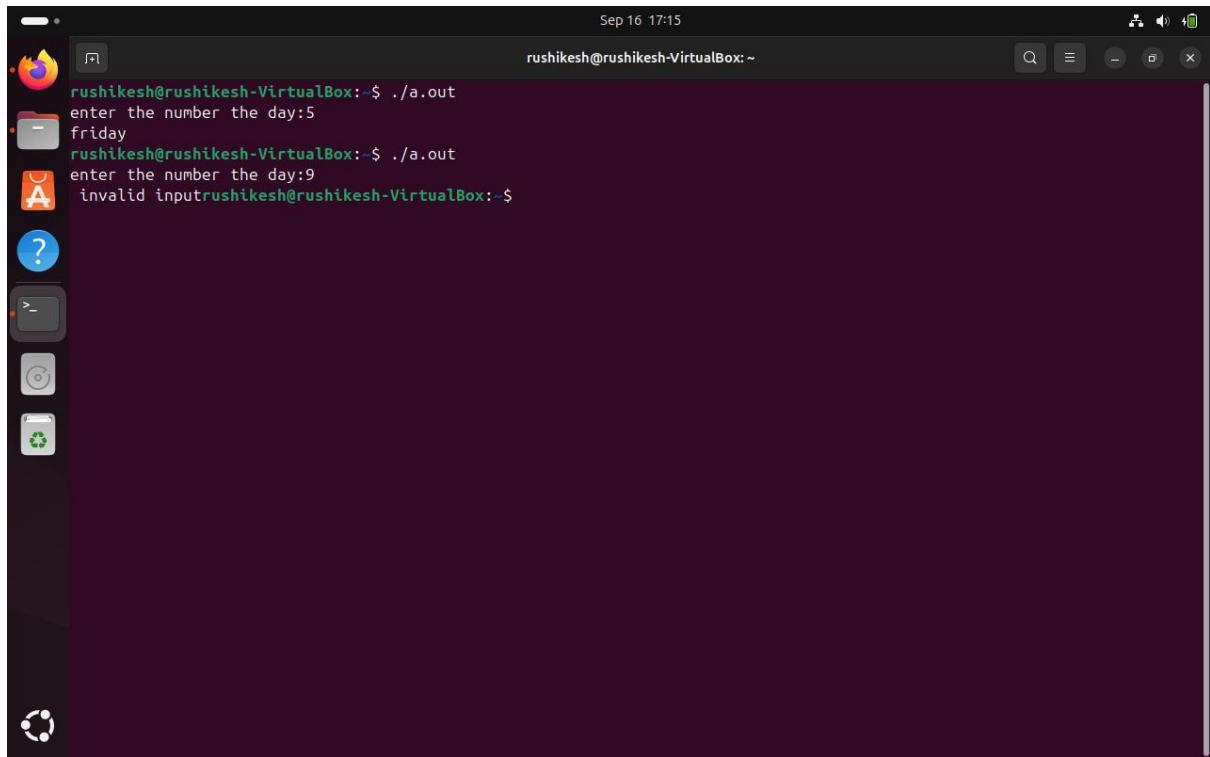
- ^G Help
- ^X Exit
- ^O Write Out
- ^R Read File
- ^W Where Is
- ^A Replace
- ^K Cut
- ^U Paste
- ^T Execute
- ^J Justify
- ^C Location
- ^/ Go To Line
- M-U Undo
- M-E Redo
- M-A Set Mark
- M-G Copy

This screenshot shows the same terminal window as the first one, but the C code now includes a return statement at the end:

```
GNU nano 7.2
scanf("%d",&n);
switch (n){
case 1:
printf("monday\n");
break;
case 2:
printf("tuesday\n");
break;
case 3:
printf("wednesday\n");
break;
case 4:
printf("thursday\n");
break;
case 5:
printf("friday\n");
break;
case 6:
printf("saturday\n");
break;
case 7:
printf("sunday\n");
default:
printf(" invalid input\n");
}
return 0;
}
```

The terminal interface and keyboard shortcuts are identical to the first screenshot.

Output:-



-----Day3-----

Q1)Find the factorial of a given number using while loop.

```
C_Assignment : vi — Konsole
File Edit View Bookmarks Settings Help
//Q2.Find the factorial number using while loop.
#include <stdio.h>

int main() {
    int num;
    int factorial = 1;

    // Get the number from the user
    printf("Enter a positive integer: ");
    scanf("%d", &num);

    // Check for non-negative integer
    if (num < 0) {
        printf("Factorial is not defined for negative numbers.\n");
        return 1; // Exit with an error code
    }

    // Calculate factorial using a while loop
    int i = num;
    while (i > 0) {
        factorial *= i;
        i--;
    }

    // Output the factorial
    printf("Factorial of %d is %d\n", num, factorial);
}

return 0;
}

"Q2.c" 32L, 629C
26,36 8:38:00 PM All
ChatGPT - Google Chrome C_Assignment : vi — Konsole
Thu,12th Sep 2024
```

Output:-

```
C_assignment : bash — Konsole
File Edit View Bookmarks Settings Help
corporate@ACTS31:~/C_Assignment> gcc Q2.c
corporate@ACTS31:~/C_Assignment> ./a.out
Enter a positive integer: 3
Factorial of 3 is 6
corporate@ACTS31:~/C_Assignment> █
```

Q.2) Write a c program to reverse a given number using while loop.

The screenshot shows a terminal window titled "C_Assignment : vi — Konsole". The code displayed is a C program to reverse a given number. The code uses a while loop to repeatedly extract the last digit of the input number and append it to a reversed number. It includes comments explaining each step. The terminal window also shows the file name "03.c", its size "25L", and the number of characters "665c".

```
File Edit View Bookmarks Settings Help
//03 Write a c program to reverse a given number
#include <stdio.h>

int main() {
    int num, reversed = 0, remainder;
    // Get the number from the user
    printf("Enter an integer: ");
    scanf("%d", &num);

    // Reverse the number
    int original_num = num; // Store the original number to display later
    while (num != 0) {
        remainder = num % 10; // Get the last digit
        reversed = reversed * 10 + remainder; // Append the digit to the reversed number
        num /= 10; // Remove the last digit from num
    }

    // Output the reversed number
    printf("Reversed number of %d is %d\n", original_num, reversed);
    return 0;
}

"03.c" 25L, 665c
12,27 8:48:49 PM
Thu, 12th Sep 2024
C.Assignment : vi — Konsole
```

Output:-

```
C_Assignment : bash — Konsole
File Edit View Bookmarks Settings Help
corporate@ACTS31:~/C_Assignment> gcc Q2.c
corporate@ACTS31:~/C_Assignment> ./a.out
Enter a positive integer: 3
Factorial of 3 is 6
corporate@ACTS31:~/C_Assignment> vi Q3.c
corporate@ACTS31:~/C_Assignment> gcc Q3.c
corporate@ACTS31:~/C_Assignment> ./a.out
Enter an integer: 67
Reversed number of 67 is 76
corporate@ACTS31:~/C_Assignment> vi Q3.c
corporate@ACTS31:~/C_Assignment> gcc Q3.c
corporate@ACTS31:~/C_Assignment> ./a.out
Enter an integer: 789
Reversed number of 789 is 987
corporate@ACTS31:~/C_Assignment>
```

Q3. Write a program to add elements of an array.

```
//Q1. Write a c program to add elements of an array
#include <stdio.h>
//main function
int main() {
    int n, i;
    int sum = 0;
    // Getting the number of elements in the array
    printf("Enter the number of elements of an array: ");
    //read input from user
    scanf( "%d", &n);
    // Declared an array of size n
    int arr[n];
    // Input the elements of the array
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    // Calculate the sum of the elements
    for(i = 0; i < n; i++) {
        sum += arr[i];
    }
    // Output the sum
    printf("The sum of the elements is: %d\n", sum);
    return 0;
}

-- INSERT --

```

1,9 All 8:06:47 PM Thu,12th Sep 2024

Output:-

```
corporate@ACTS31:~> mkdir C_Assignment
corporate@ACTS31:~> cd C_Assignment
corporate@ACTS31:~/C_Assignment> vi Q1.c
corporate@ACTS31:~/C_Assignment> gcc Q1.c
corporate@ACTS31:~/C_Assignment> ./a.out
Enter the number of elements of an array: 3
Enter 3 elements:
4
5
6
The sum of the elements is: 15
corporate@ACTS31:~/C_Assignment> vi Q1.c
corporate@ACTS31:~/C_Assignment> gcc Q1.c
corporate@ACTS31:~/C_Assignment> ./a.out
Enter the number of elements of an array: 3
Enter 3 elements:
2
4
5
The sum of the elements is: 11
corporate@ACTS31:~/C_Assignment>
```

8:08:50 PM Thu,12th Sep 2024

Q4. Write a c program using while loop and accept input from user until user enter multiple of 10.

The screenshot shows a code editor window with a dark theme. The menu bar includes File, Edit, View, Bookmarks, Settings, and Help. The code itself is a C program:

```
#include<stdio.h>
#include <math.h>
//main function starts here
int main()
{
    //declare two variables
    int a=1;
    int n;
    while(a!=0)
    {
        //taking input number from user
        scanf("%d",&n);
        //if the number is divisible by 10 means we have to stop here
        if(n%10==0)
            {a=0;}
        else
            //otherwise print the number
            printf("%d\n",n);
    }
    return 0;
}
```

Output:-

```
File Edit View Bookmarks Settings Help
corporate@ACTS05:~> gcc cassingment.c
corporate@ACTS05:~> ./a.out
7
7
8
8
2
2
3
3
8
8
9
9
10
corporate@ACTS05:~>
```

5) Write a program to calculate the average age of employees from the array.

```
File Edit View Bookmarks Settings Help
#include<stdio.h>
//main function starts here
int main()
{
    //initializing an array
    int arr[]={24,35,16,18,22,12};
    //calculating size of array
    int length=sizeof(arr)/sizeof(arr[0]);
    //declaring sum variable
    int sum=0;
    for(int i =0;i<length;i++)
    {
        sum=sum+arr[i];
    }
    //calculating the average
    int n =(int) sum/length;
    //printing the average
    printf("%d\n",n);
    return 0;
}
```

Output:-

```
File Edit View Bookmarks Settings Help
corporate@ACTS05:~> gcc cassingment.c
corporate@ACTS05:~> ./a.out
21
corporate@ACTS05:~>
```

6) Write a program to calculate the youngest among the employees.

```
#include<stdio.h>
//main function starts here
int main()
{
    //declare array
    int arr[]={56,80,72,12,90,76};
    //declared one minimum variable
    int min=arr[0];
    //calculating size of array
    int length=sizeof(arr)/sizeof(arr[0]);
    //for loop for finding array
    for(int i =0;i<length;i++)
    {
        if(min>arr[i])
            min=arr[i];
    }
    //finally we got minimum variable
    printf("%d\n",min);
    return 0;
}
```

Output:-

```
File Edit View Bookmarks Settings Help
corporate@ACTS05:~> gcc cassingment.c
corporate@ACTS05:~> ./a.out
12
corporate@ACTS05:~>
```

7) Write a program to demonstrate arrays of the following types float,char,bool.

```
File Edit View Bookmarks Settings Help
#include<stdio.h>
//main function starts here
int main()
{//making array of float type
    float arr[]={1.2,1.3,1.4,1.5};
    int length=sizeof(arr)/sizeof(arr[0]);
    printf("Float type arr\n");
    for(int i =0;i<length;i++)
    {
        printf("%f\n",arr[i]);
    }
    printf("\n");
    //making array of char type
    char brr[]={'a','b','c','d','e','f'};
    int len =sizeof(brr)/sizeof(brr[0]);
    printf("Character type arr\n");
    for(int i=0;i<len;i++)
    {
        printf("%c\n",brr[i]);
    }
    printf("\n");
    //make array of bool type
    int curr[]={1,0,1,0,1,1,1};
    int size=sizeof(curr)/sizeof(curr[0]);
    printf("Boolean type arr\n");
    for(int i =0;i<size;i++)
    {
        if(curr[i])
            printf("true\n");
        else
            printf("false\n");
    }
    printf("\n");
    return 0;
}
```

Output:-

```
corporate@ACTS05:~> gcc cassingment.c
corporate@ACTS05:~> ./a.out
Float type arr
1.200000
1.300000
1.400000
1.500000

Character type arr
a
b
c
d
e
f

Boolean type arr
true
false
true
false
true
true
true
true

corporate@ACTS05:~>
```

8)Write a program to calculate the total cost on a purchase order with the following details. There are 20 items costing 11.2 per item. Display the total cost in dollars.

```
#include<stdio.h>
//main function starts here
int main()
{
    //here we are calculating price of given items
    int items = 20;
    float price=11.2;
    //calculating amount
    float amount=items*price;
    //printing price in dollars
    printf("Total price: %.2f",amount);
}
```

Output:-

```
corporate@ACTS05:~> gcc cassingment.c
corporate@ACTS05:~> ./a.out
Total price: $224.00corporate@ACTS05:~>
```

Q1>Write a program to check a given input is palindrome.

The screenshot shows a terminal window titled "GNU nano 7.2" with the command "palindrome.c" open. The code is as follows:

```
#include<stdio.h>
#include<string.h>
int main(){
    char str[100];
    int start,end,ispalindrome;
    printf("enter a string:");
    scanf("%s",str);
    int length= strlen(str);
    start=0;
    end=length-1;
    ispalindrome=1;
    while (start<end){
        if (str[start]!=str [end]){
            ispalindrome = 0;
            break;
        }
        start++;
        end--;
    }
    if (ispalindrome){
        printf("the string is a palindrome\n");
    }else{
        printf("the string is not a palindrome.\n");
    }
    return 0;
}
```

At the bottom, there are various keyboard shortcuts: Help (^G), Exit (^X), Write Out (^O), Read File (^R), Where Is (^W), Replace (^R), Cut (^K), Paste (^U), Execute (^T), Justify (^J), Location (^C), Go To Line (^L), Undo (^U), Redo (^E), Set Mark (^A), and Copy (^M).

Output:-

The screenshot shows a terminal window with the following session:

```
rushikesh@rushikesh-VirtualBox:~$ nano palindrome.c
rushikesh@rushikesh-VirtualBox:~$ gcc palindrome.c
palindrome.c: In function 'main':
palindrome.c:13:16: error: expected expression before '[' token
   13 | if(str[start]!=end]){
      |
palindrome.c:14:1: error: 'ispalindrome' undeclared (first use in this function); did you mean 'ispalindrome'?
  14 | ispalindrome = 0;
     | ^
     | ispalindrome
palindrome.c:14:1: note: each undeclared identifier is reported only once for each function it appears in
rushikesh@rushikesh-VirtualBox:~$ nano string.c
rushikesh@rushikesh-VirtualBox:~$ nano palindrome.c
rushikesh@rushikesh-VirtualBox:~$ gcc palindrome.c
palindrome.c: In function 'main':
palindrome.c:13:16: error: expected expression before '[' token
   13 | if(str[start]!=end){
      |
rushikesh@rushikesh-VirtualBox:~$ nano palindrome.c
rushikesh@rushikesh-VirtualBox:~$ gcc palindrome.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter a string:rushikesh
the string is not a palindrome.
rushikesh@rushikesh-VirtualBox:~$ nano palindrome.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter a string:rushikesh
the string is not a palindrome.
rushikesh@rushikesh-VirtualBox:~$ nano palindrome.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter a string:madam
the string is a palindrome
rushikesh@rushikesh-VirtualBox:~$
```

Q2>Write a program to check whether two strings are equal to or not.

The screenshot shows a terminal window titled "rushikesh@rushikesh-VirtualBox: ~". The window contains a nano editor session with the following C code:

```
GNU nano 7.2
#include<stdio.h>
#include<string.h>
int main(){
char str1[100],str2[100];
printf("enter the first string :");
scanf("%s",str1);
printf("enter the second string:");
scanf("%s",str2);
// compare the string
if(strcmp(str1,str2)==0){
printf("the string are equal\n");
}else{
printf("the string are not equal.\n");
}
return 0;
}
```

The terminal window has a dark theme and includes standard nano keybindings at the bottom.

Output:-

The screenshot shows a terminal window titled "rushikesh@rushikesh-VirtualBox: ~". The session starts with the user navigating to the directory and compiling the C program:

```
rushikesh@rushikesh-VirtualBox:~$ nano string.c
rushikesh@rushikesh-VirtualBox:~$ gcc string.c
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/13/../../../../x86_64-linux-gnu/Srct1.o: in function '_start':
(.text+0x1b): undefined reference to 'main'
collect2: error: ld returned 1 exit status
```

After fixing the main() declaration and recompiling, the user runs the program:

```
rushikesh@rushikesh-VirtualBox:~$ nano string.c
rushikesh@rushikesh-VirtualBox:~$ gcc string.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter the first string:rushikesh
enter the second string:sonekar
the string are not equal.
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter the first string:rushikesh
enter the second string:rushikesh
the string are equal
rushikesh@rushikesh-VirtualBox:~$
```

Q3>Write a program to create reverse string.

The screenshot shows a terminal window titled "rushikesh@rushikesh-VirtualBox: ~". The file being edited is "reverse.c". The code implements a function to reverse a string using a two-pointer approach. The terminal window includes a toolbar with icons for file operations like Open, Save, and Cut, and a menu bar with keyboard shortcuts for various functions.

```
GNU nano 7.2
#include<stdio.h>
void reversestring(char str[]);
int main(){
char str[100];
printf("enter a string:");
scanf("%s",str);
reversestring(str);
printf("reversed string:%s\n",str);
return 0;
}
void reversestring(char str[]){
int start =0;
int end =0;
while (str[end] != '\0'){
end++;
}
end--;
while (start < end){
char temp = str[start];
str[start] = str[end];
str[end] = temp;
start++;
end--;
}
}
```

[Read 25 lines]

Help Exit Write Out Read File Replace Where Is Cut Paste Execute Justify Location Go To Line Undo Redo Set Mark

Output:-

The screenshot shows a terminal window titled "rushikesh@rushikesh-VirtualBox: ~". The user runs the compiled program "a.out" and enters the string "rushikesh". The output shows the reversed string "hsekihsur". The terminal window has a similar interface to the one above, with a toolbar and a menu bar.

```
rushikesh@rushikesh-VirtualBox:~$ nano reverse.c
rushikesh@rushikesh-VirtualBox:~$ gcc reverse.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter a string:rushikesh
reversed string:hsekihsur
rushikesh@rushikesh-VirtualBox:~$
```

Q4>Write a program for addition without recursion and with recursion.

A screenshot of a terminal window titled "Windows Terminal" on a Windows 10 desktop. The terminal shows a C program for addition. The code defines two functions: `add_without_recursion` and `add_with_recursion`, and a main function that prompts the user for two numbers and a choice of method. The terminal window has a dark theme. Below the terminal is a standard Windows taskbar with icons for File Explorer, Edge browser, File History, Task View, Start, Taskbar settings, and system status indicators like battery level and signal strength.

Output:-

Q5.>Write a program for Fibonacci series and factorial using recursion.

The screenshot shows a terminal window titled "factorial.c" with the following C code:

```
GNU nano 7.2
#include<stdio.h>
int fibonacci(int n){
    if(n<=1){
        return n;
    }
    return fibonacci(n-1)+fibonacci(n-2);
}
int factorial (int n){
    if(n==0){
        return 1;
    }
    return n*factorial(n-1);
}
int main(){
    int n;
    printf("enter a number to calculate its fibonacci number:\n");
    scanf("%d",&n);
    if(n<0){
        printf("fibonacci is not defined for negative number:\n");
    }
    else{
        printf("fibonacci number at a position %d is %d .\n",n,fibonacci(n));
    }
    printf("enter the number to calculate its factorial:\n");
    scanf("%d",&n);
    if(n<0){
        printf("factorial is not defined for negative number:\n");
    }
}
```

Below the code, there is a toolbar with various icons and keyboard shortcuts for file operations like Help, Exit, Write Out, Read File, Replace, Cut, Paste, Execute, Location, Go To Line, Undo, Redo, Set Mark, and a status bar indicating "Read 33 lines".

Output:-

The screenshot shows a terminal window with the following session:

```
rushikesh@rushikesh-VirtualBox:~$ nano factorial.c
rushikesh@rushikesh-VirtualBox:~$ gcc factorial.c
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter a number to calculate its fibonacci number:
20
fibonacci number at a position 20 is 6765 .
enter the number to calculate its factorial:
78
factorial of 78 is0.
rushikesh@rushikesh-VirtualBox:~$ ./a.out
enter a number to calculate its fibonacci number:
10
fibonacci number at a position 10 is 55 .
enter the number to calculate its factorial:
20
factorial of 20 is-2102132736.
rushikesh@rushikesh-VirtualBox:~$
```