**Introduction to C language**

**LAB # 05**

****

**Spring 2021**

**CSE204L Operating System**

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr. Mian Ibad Ali Shah**

June 3 , 2021

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Exercise Tasks**

1. Run all the above example problems and try to understand the concepts.

**Basic syntax of C Programming**

**Source code:**

#include <stdio.h>

int main()

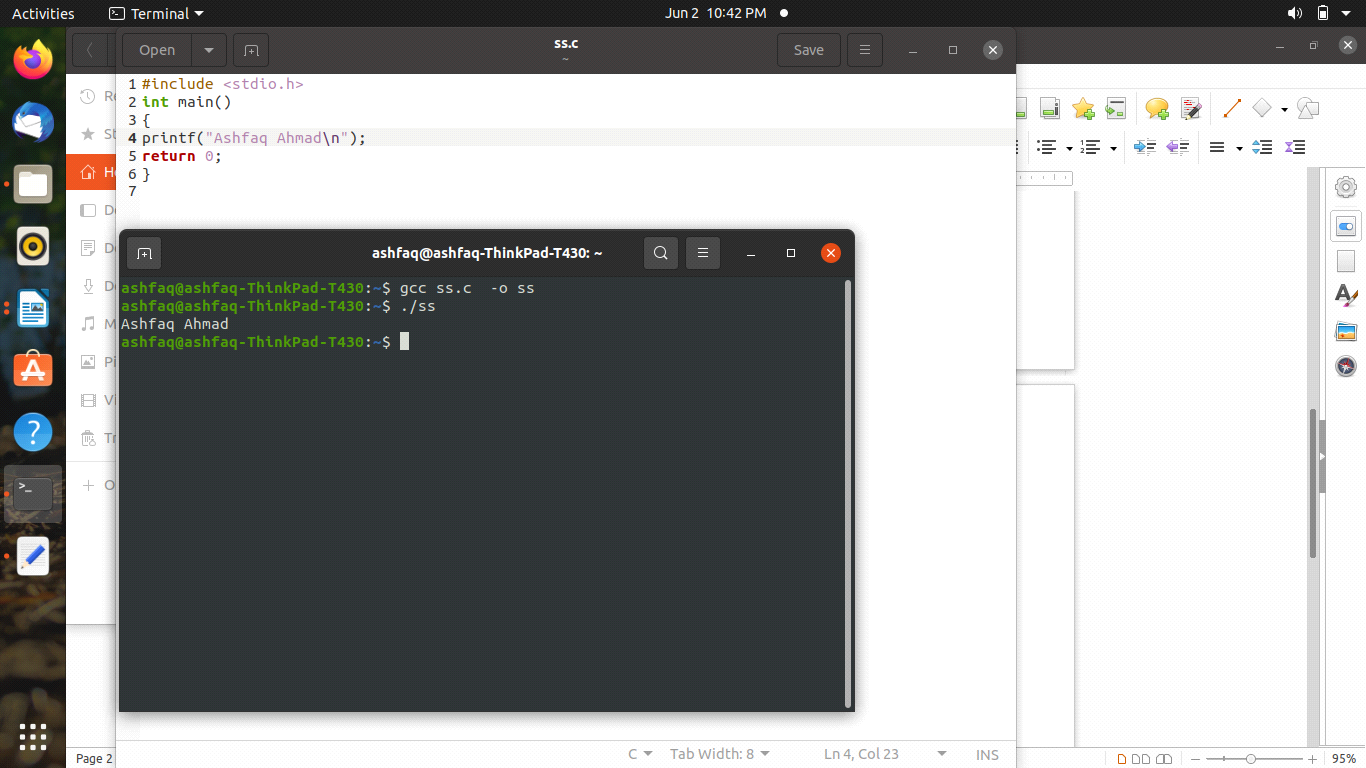
{

printf("Ashfaq Ahmad\n");

return 0;

}

**output:**



**A Simple C program with more than one function (Parameters passed by value)**

**Source code:**

#include <stdio.h>

int add(int x,int y)

{

return x+y;

}

int product(int x,int y)

{

return x\*y;

}

int main()

{

int a,b;

printf("please enter the value of x and y\n");

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

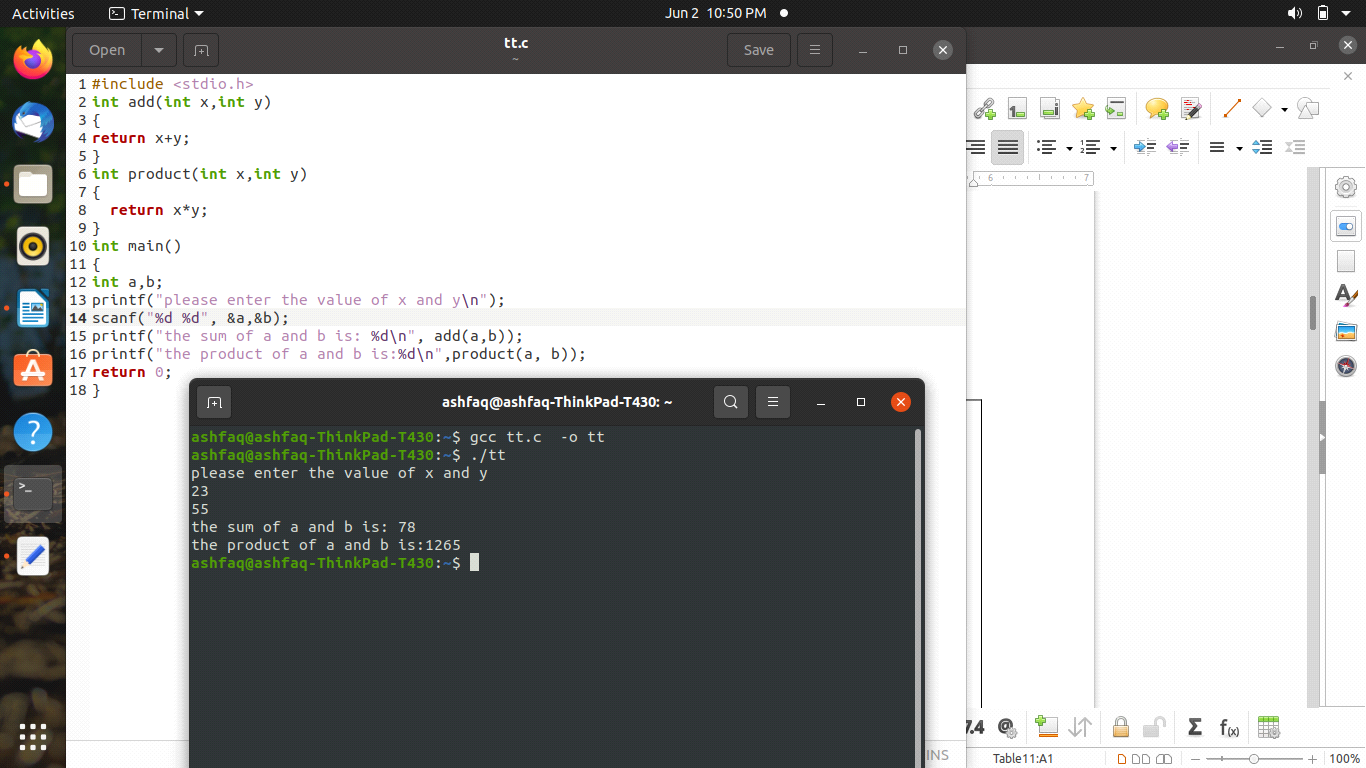
printf("the sum of a and b is: %d\n", add(a,b));

printf("the product of a and b is:%d\n",product(a, b));

return 0;

}

**Output:**



**Basic concepts of Pointers in C**

**Source code:**

#include <stdio.h>

int main()

{

int a;

int \*p;

printf("please enter the value of a: \n");

scanf("%d", &a);

p=&a;

printf("the value of a is: %d\n",a);

printf("the address of a is: %x\n",&a);

printf("the value of p is: %x\n",p);

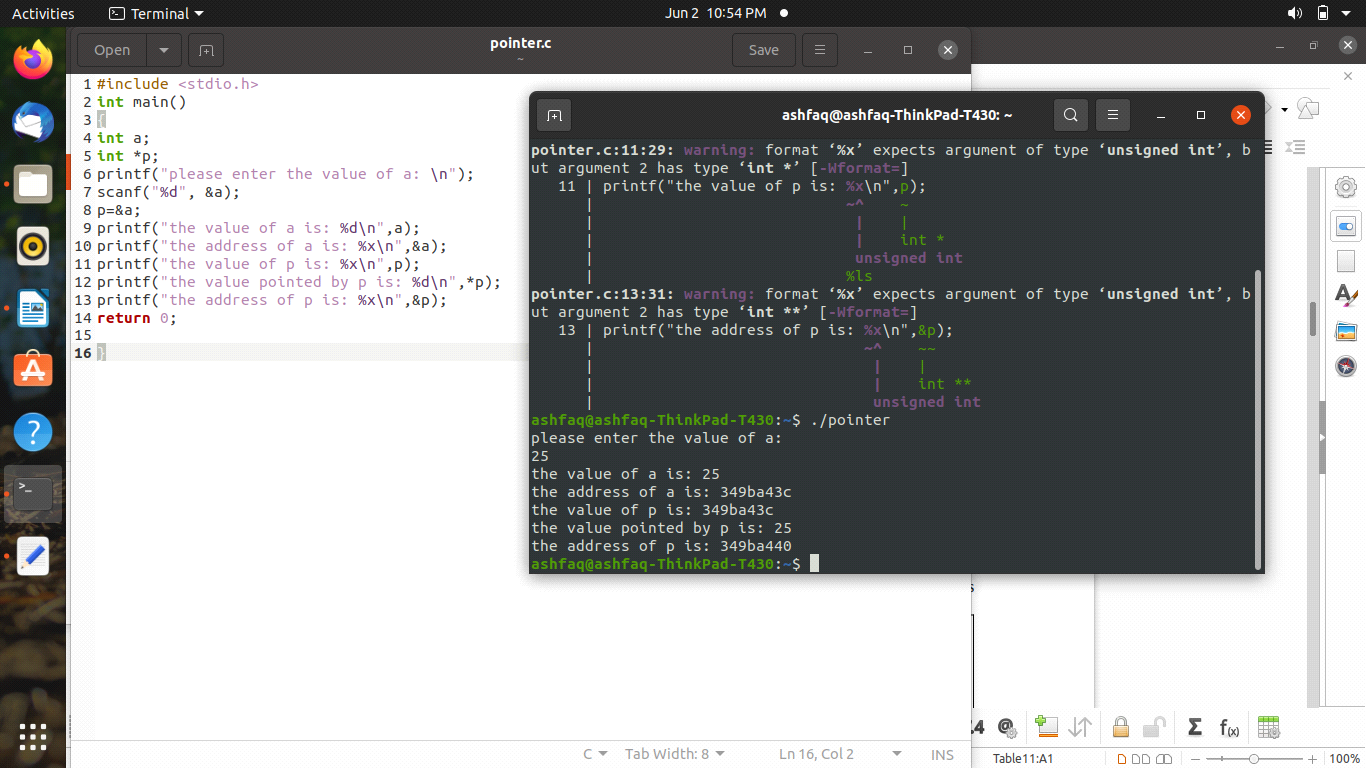
printf("the value pointed by p is: %d\n",\*p);

printf("the address of p is: %x\n",&p);

return 0;

}

**Output:**



**Passing parameters to function by pointers**

#include <stdio.h>

void add(int a,int b,int \*p)

{

\*p=a+b;

}

int main()

{

int x,y,z;

printf("please enter the values of x and y: \n");

scanf("%d %d", &x,&y);

add(x,y,&z);

printf("the value of sum is: %d\n",z);

return 0;

}

**Using Arrays in C**

**Source code:**

#include <stdio.h>

//#define size 5

//we can also define the size of array like above method or we can also put it in run time like below.

int main()

{

int size;

printf("please enter the size of array: \n");

scanf("%d",&size);

int x[size],y[size],z[size];

printf("please enter %d elements in first array: \n",size);

for(int i=1; i<=size; i++)

{

scanf("%d",&x[i]);

}

printf("please enter %d elements in 2nd array: \n",size);

for(int i=1; i<=size; i++)

{

scanf("%d",&y[i]);

}

printf("now the subtraction of first array from 2nd array is: \n");

for(int i=1; i<=size; i++)

{

z[i]=y[i]-x[i];

printf("\n %d",z[i]);

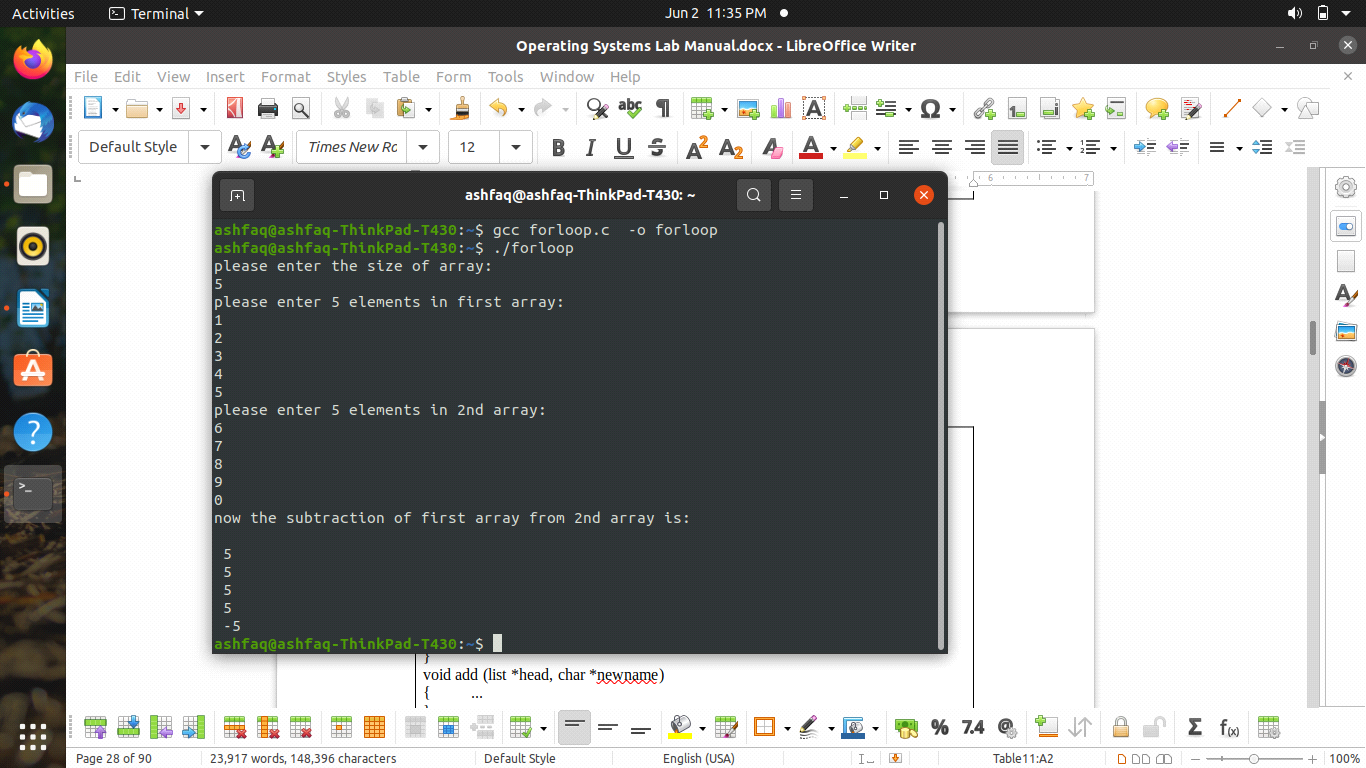
}

printf("\n");

return 0;

}

Output:



**Using Structures in C**

**Source code:**

#include <stdio.h>

struct student

{

char name[20];

int reg\_no;

};

int main()

{

struct student a1,a2;

printf("please enter the student name and reg no: \n");

scanf("%s %d",&a1.name,&a1.reg\_no);

printf("the name is: %s\n",a1.name);

printf("And reg no is: %d\n",a1.reg\_no);

printf("please enter the student name and reg no: \n");

scanf("%s %d",&a2.name,&a2.reg\_no);

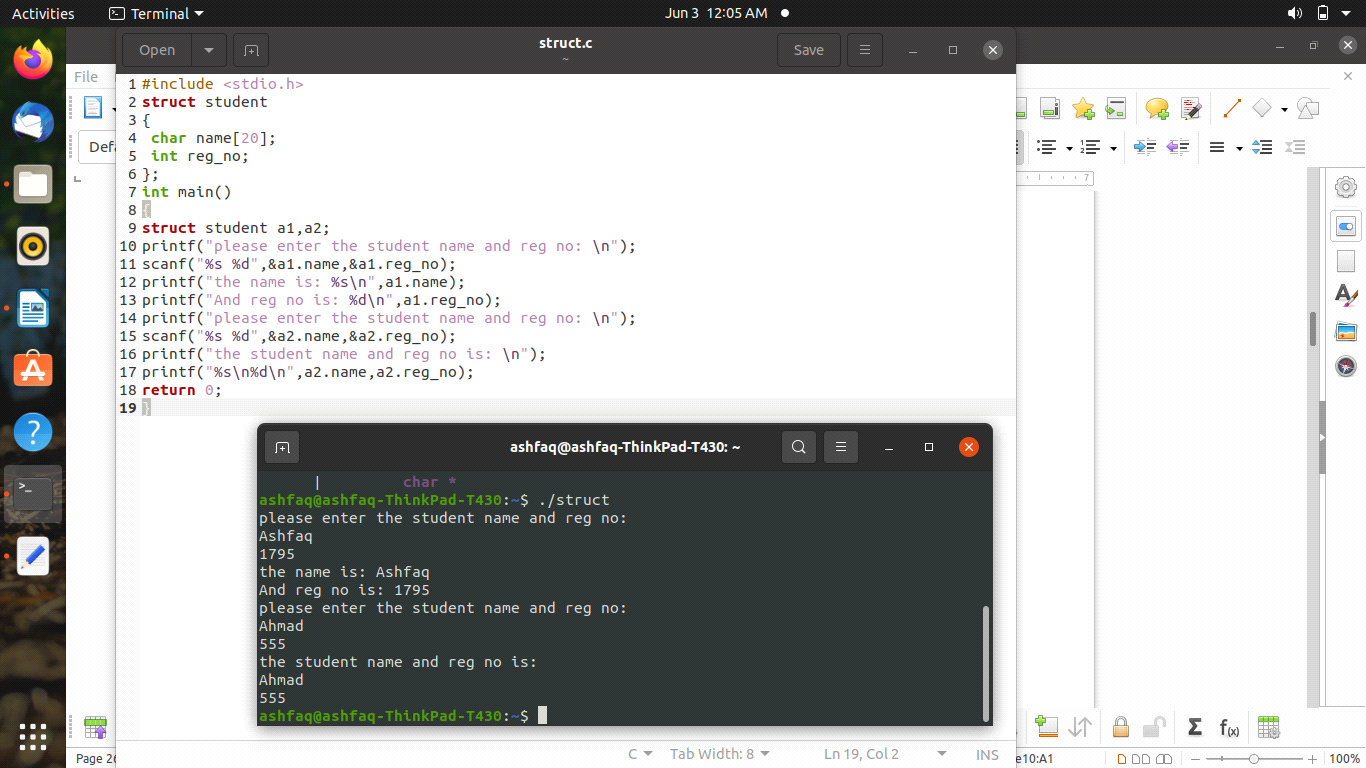
printf("the student name and reg no is: \n");

printf("%s\n%d\n",a2.name,a2.reg\_no);

return 0;

}

**Output:**



**Task no 2**

1. Write a complete menu driven program to do the following:

* Build a linked list to save a list of names. Name will not exceed 50 characters.
* Write a function add to append a new name to the list. The function prototype is given as

void add (list \*head, char \*newname);

* Write a function search to look for a given name in the list. If that name is found in list then the function should return true, otherwise, return false.
* Write a main method to test your two functions.

            In C language, the boolean type and the boolean literals (true, false) are not defined. We can define these in our program as follow:

typedef enum {false = 0, true} boolean;

             The skeleton of your program should look like the following:

|  |
| --- |
| Program  #include <stdio.h>  #include <stdlib.h>  typedefstructlist {  ...  ...  } list;  typedefenum{false=0, true} boolean;  voidadd (list \*, char \*);  booleansearch (list \*, char \*);  intmain()  { ...  ...  }  voidadd (list \*head, char \*newname)  { ...  }  booleansearch (list \*head, char \*name)  { ...  } |

**Use of Linked List in C**

**Source code:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct list {

char arr[50];

struct list \* elem;

};

void add(struct list \* f, char \* str);

int search(struct list \* f, char \* str);

void main() {

struct list first;

char arr[50] = "Aamir Ibrahim";

strcpy(arr, first.arr);

first.elem = 0;

int i = 0;

for(;i < 5;i++) {

printf("Enter a name to add: ");

scanf("%s", arr);

add(&first, arr);

}

printf("Enter a name to search for: ");

scanf("%s", arr);

if(search(&first, arr) == 1) printf("Found");

else

printf("Not found");

}

void add(struct list \* f, char \* str) {

while(((\*f).elem) != 0) f = (\*f).elem;

(\*f).elem = malloc(sizeof(struct list));

strcpy(((\*(\*f).elem).arr), str);

(\*(\*f).elem).elem = 0;

}

int search(struct list \* f, char \* str) {

while(((\*f).elem) != 0) {

if(strcmp(((\*(\*f).elem).arr), str) == 0) return 1;

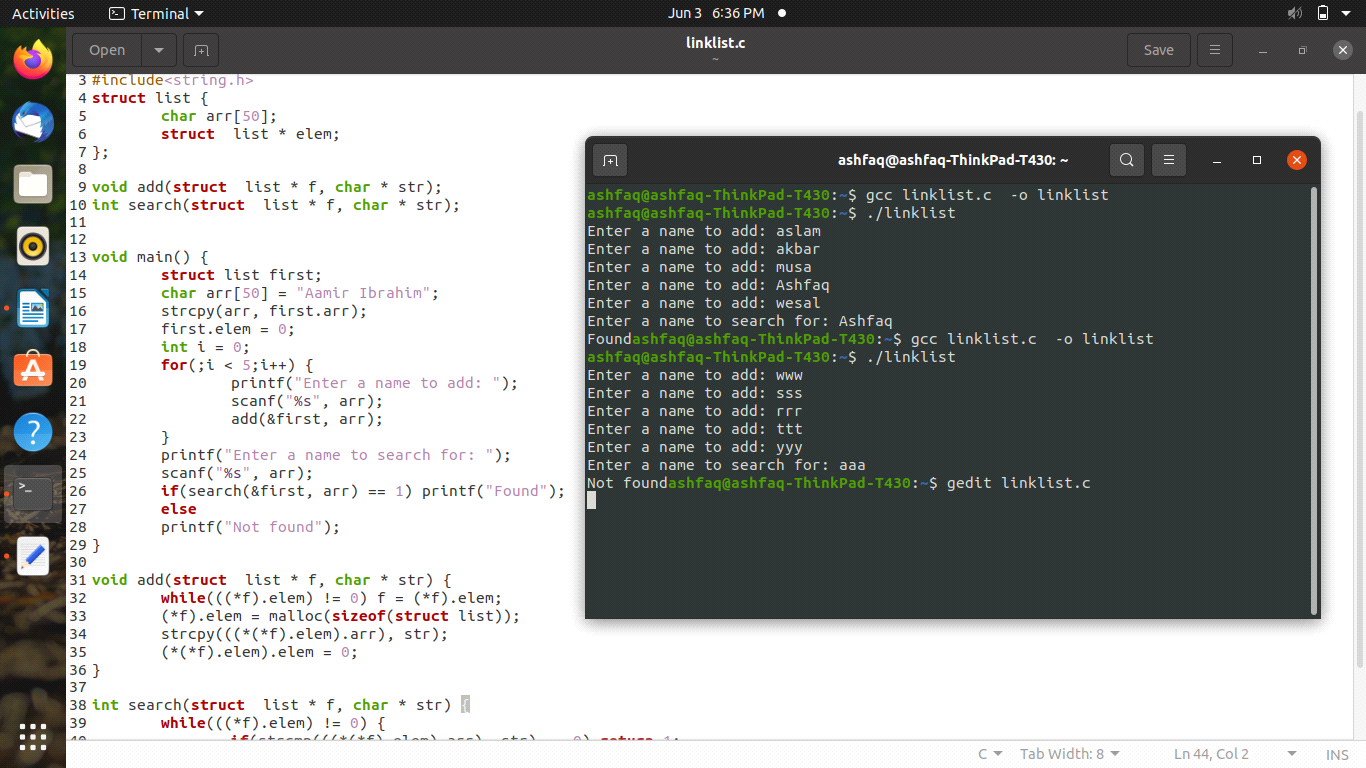
f = (\*f).elem;

}

return 0;

}

**Output:**



**THE END**