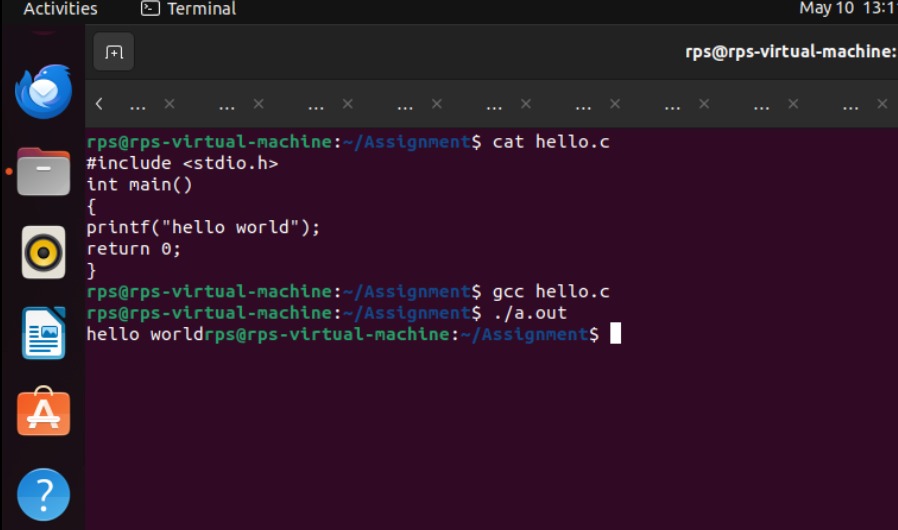
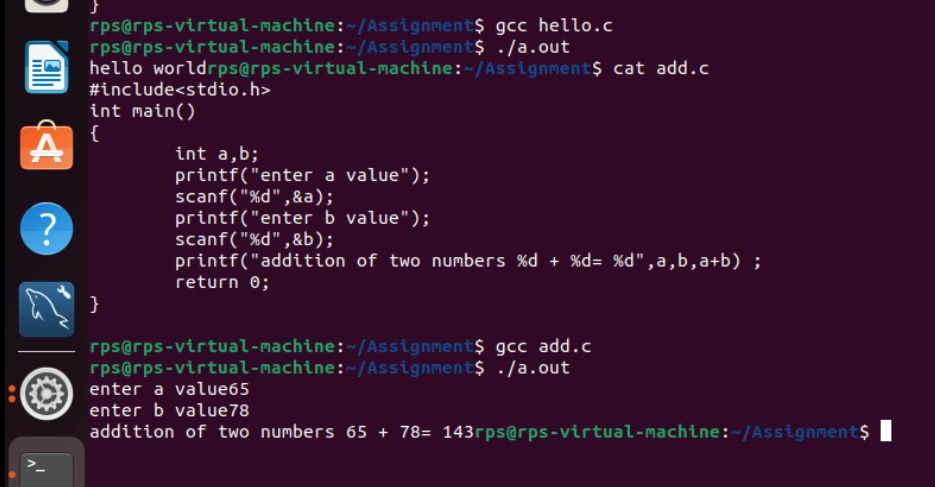
C Programs:

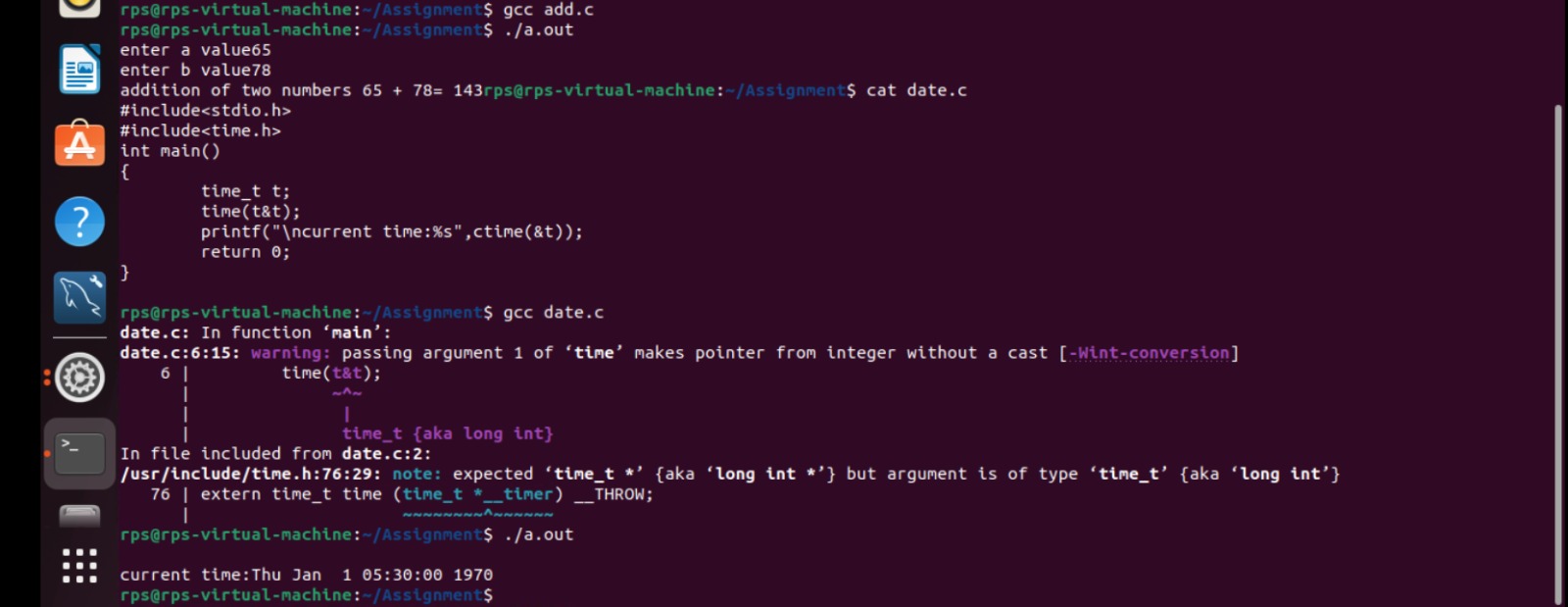
1) Write a c program to print hello world



2) Write a c program to print adding of two numbers



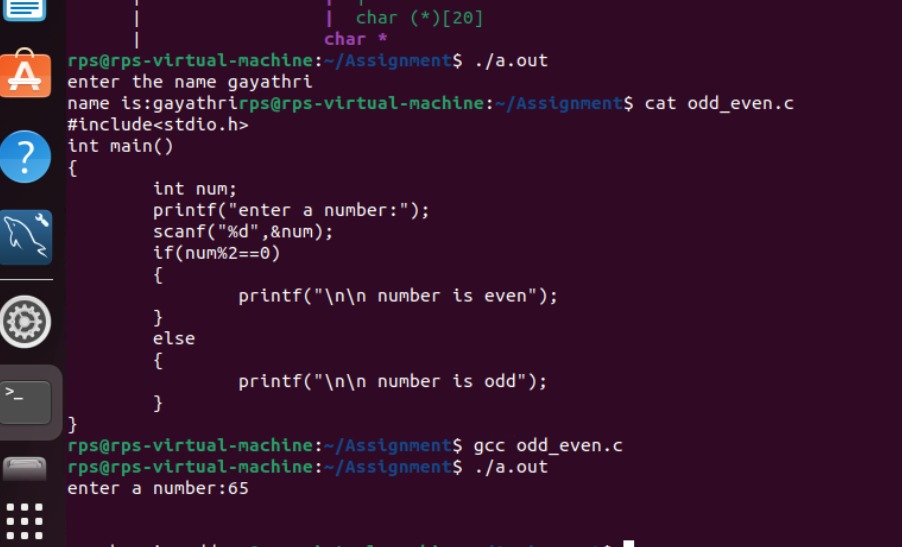
3)Write a c program to print date and time



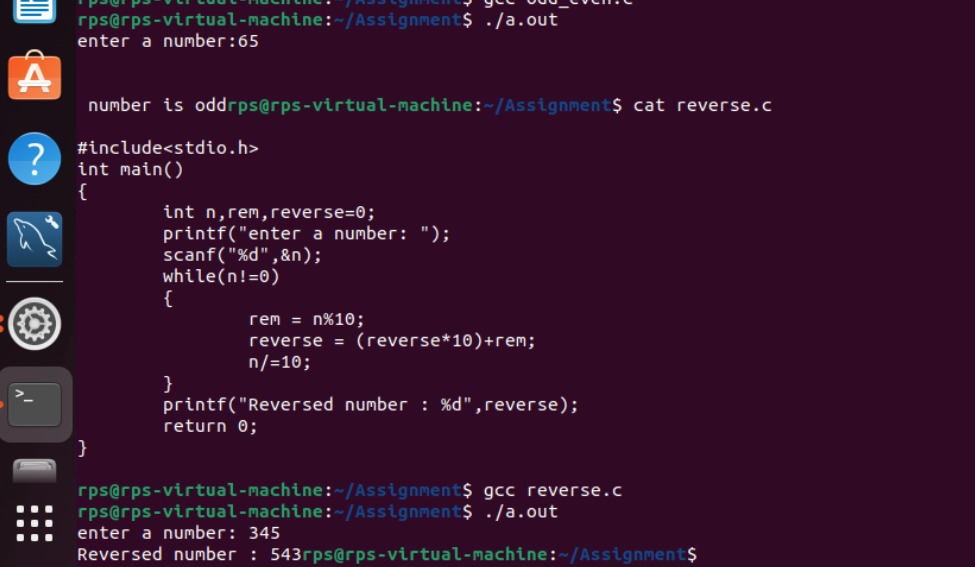
4)Write a c program to take input from user to enter their name on the screen

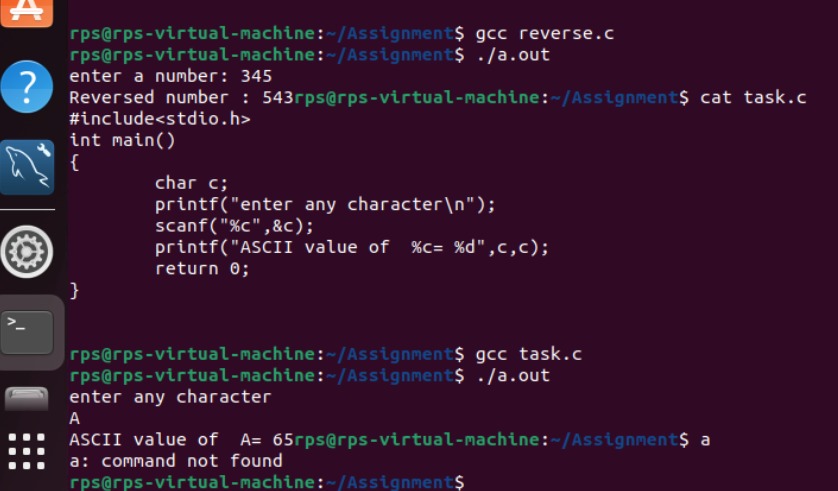


5)Write a c program to find a given number is even or odd

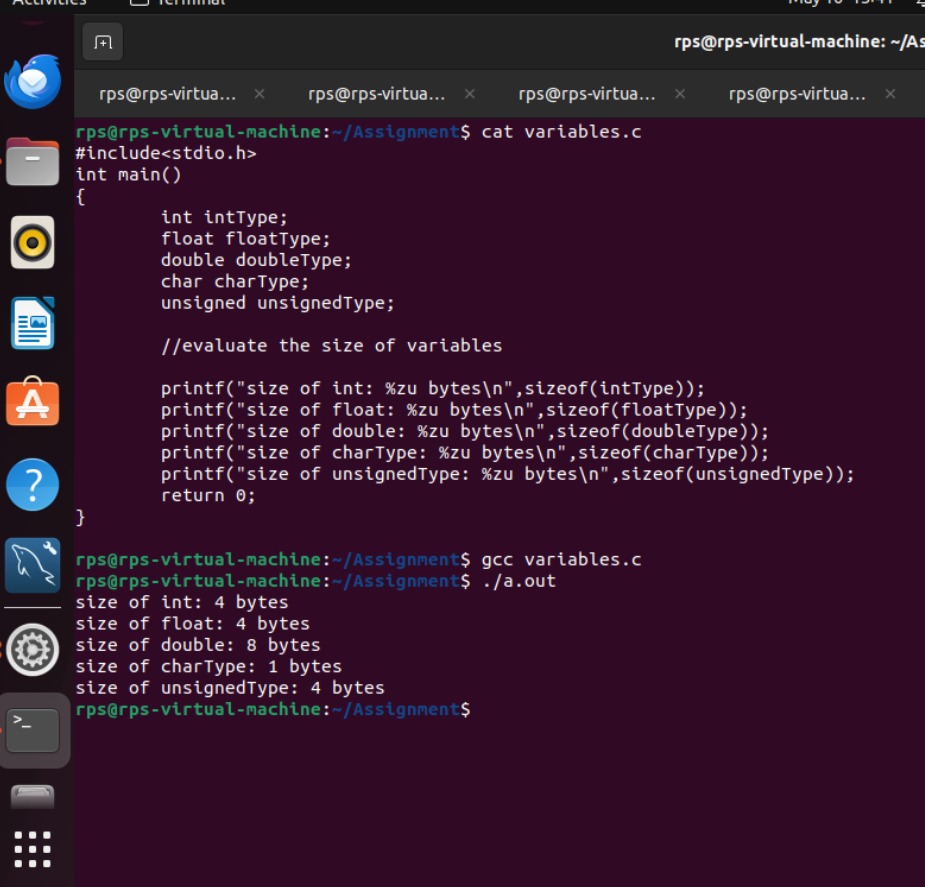


6)Write a c program to print reverse num

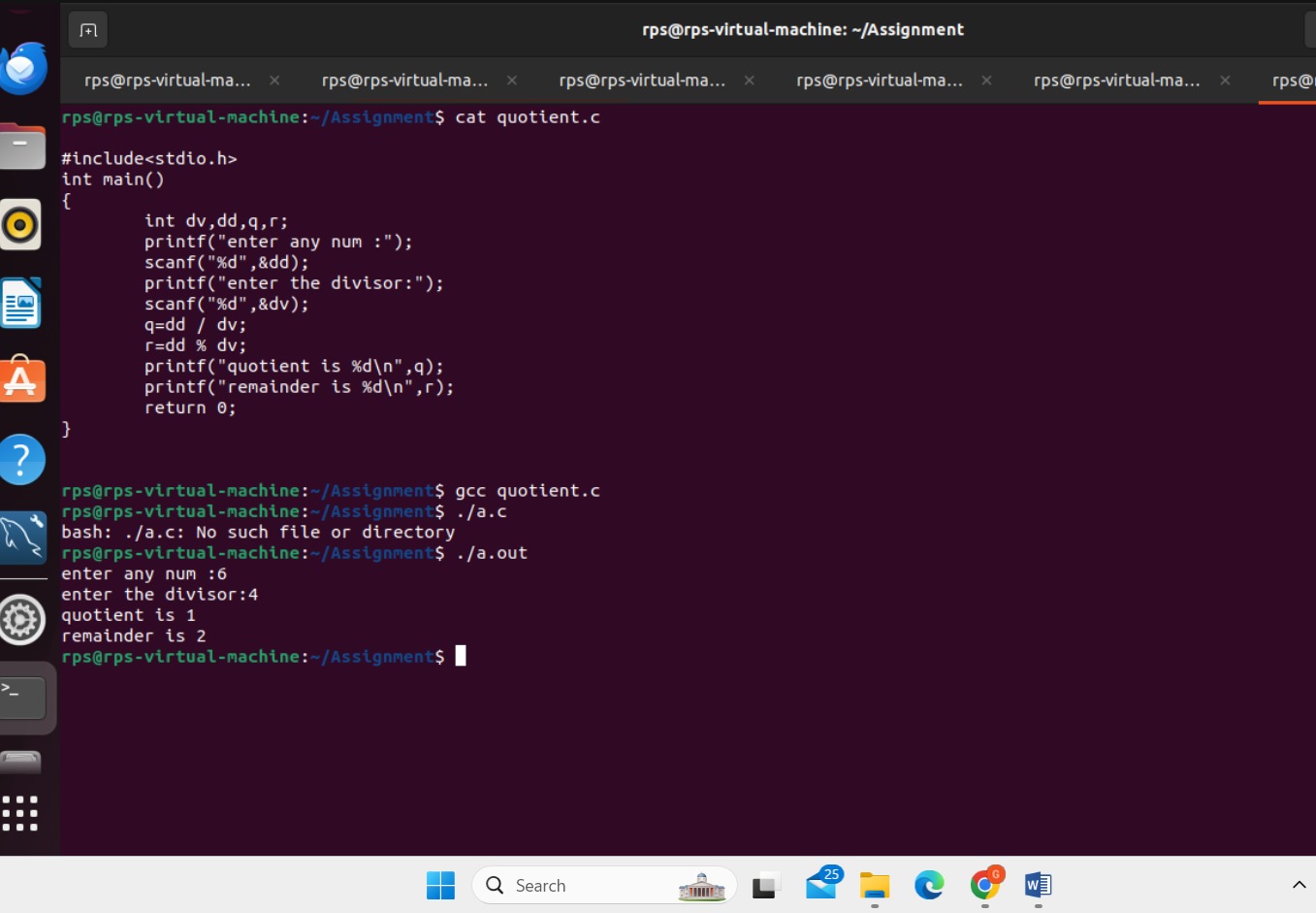


7) Printing ASCII value for a character.

8)Write a c program to find the size of variables



9)Write a c program to find quotient,remainder,dd,dv



10)write a c program to print Fibonacci series

#include<st0dio.h>

int main()

{

int n1=0,n2=1,n3,i,number;

printf("Enter the number of elements:");

scanf("%d",&number);

printf("\n %d %d",n1,n2);

for(i=2;i<number;i++)

{

n3=n1+n2;

printf("%d",n3);

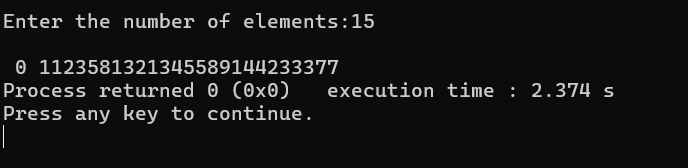
n1=n2;

n2=n3;

}

return 0;

}



11)write a c program to print factorial of a number

#include<stdio.h>

int main()

{

int i,fact=1,number;

printf("enter a number:");

scanf("%d",&number);

for(i=1;i<=number;i++)

{

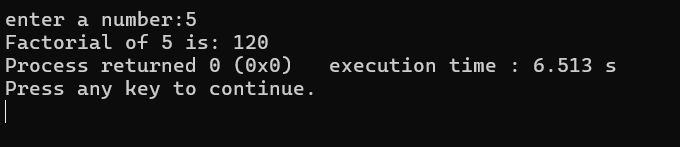
fact=fact\*i;

}

printf("Factorial of %d is: %d",number,fact);

return 0;

}



12)write a program a[20],take the input from user and print all the elements pf a[20] with their index values

#include<stdio.h>

int main()

{

int a[20];

int i;

printf("Enter 20 elements : \n");

for(i=0;i<20;i++)

{

printf("enter elements %d : ",i);

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

}

printf("\n elements of a[20]with index values:");

for(i=0;i<20;i++)

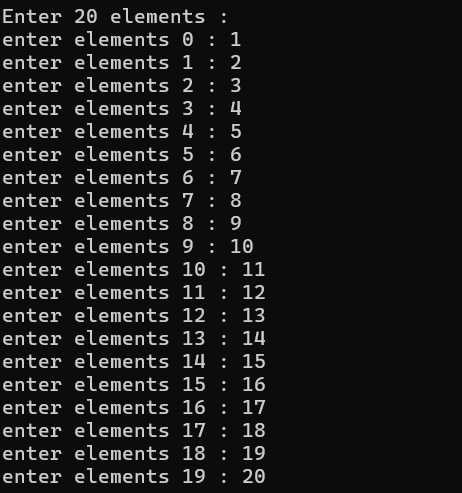
{

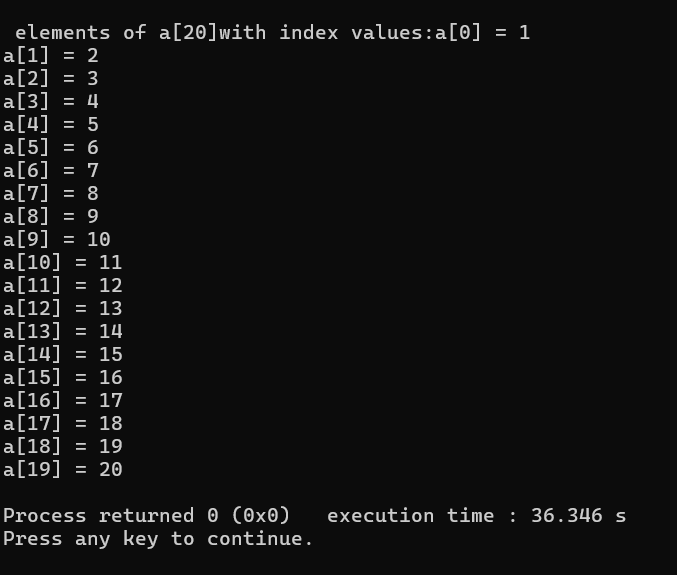
printf("a[%d] = %d\n",i,a[i]);

}

return 0;

}





13)write a program to delete a particular elements from an array

#include<stdio.h>

int main()

{

int a[20],i,j;

printf("enter the elements of the array of 20 :\n");

for(i=0;i<20;i++)

{

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

}

printf("\n enter the index of value which is to be deleted : ");

scanf("%d",&j);

a[j]=0;

printf("\n printing the elements of the array after deleting the element ");

for(i=0;i<20;i++)

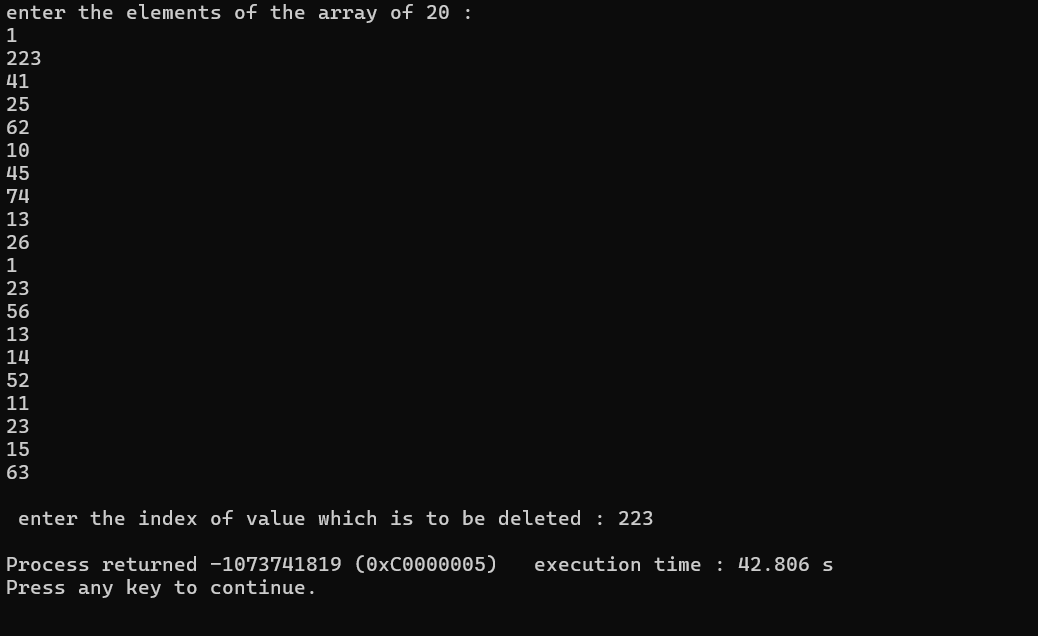
{

printf("%d",a[i]);

}

return 0;

}



14)find if there r any duplicates in a[20]

#include<stdio.h>

int main()

{

int a[20],i,j;

printf("enter the element of the array of 20 : \n ");

for(i=0;i<20;i++)

{

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

}

printf("\n printing index values of duplicate elements:\n");

for(i=0;i<20;i++)

{

for(j=i+1;j<20;j++)

{

if(a[i]==a[j])

{

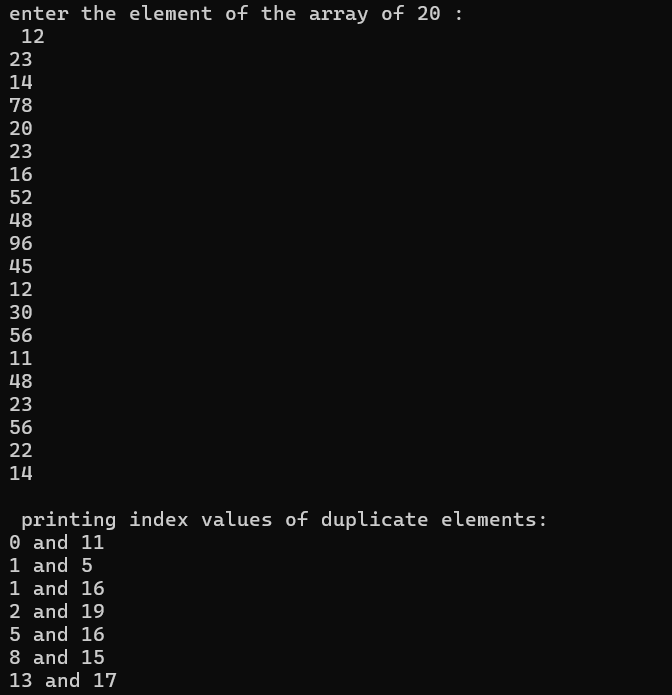
printf("%d and %d \n",i,j);

}

}

}

}



15)write a program to search [10] from a[20]

#include<stdio.h>

int main()

{

int a[20];

int i,num;

int found=0;

printf("Enter 20 elements : \n");

for(i=0;i<20;i++)

{

printf("enter elements %d : ",i);

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

}

printf("enter the elements to search:");

scanf("%d",num);

for(i=0;i<10;i++)

{

if(a[i] == num )

{

found =1;

break;

}

}

if(found)

{

printf("element %d found in a[10]\n",num);

}

else

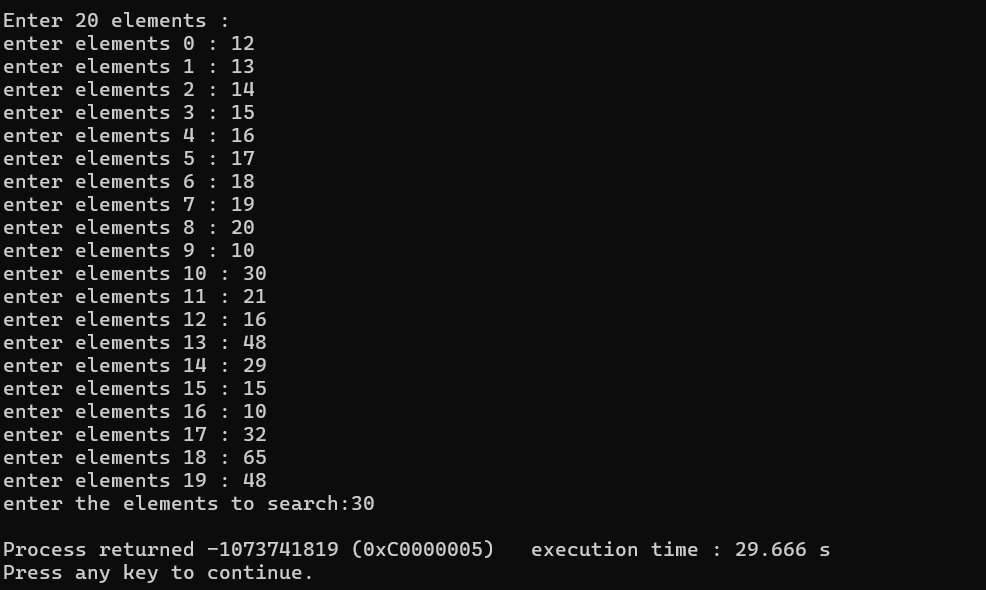
{

printf("element %d not found in a[10] \n",num);

}

return 0;

}



16)calendar:

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

main ()

{

int choice, a, b, result=0,y, rem;

start:

printf("\nWelcome to my calculator\n");

printf("\n 1. Add numbers");

printf("\n 2. Subtract numbers");

printf("\n 3. Multiply numbers");

printf("\n 4. Divide Numbers");

printf("\n 5. Exit");

printf("\nEnter your choice:: ");

scanf("%d", &choice);

switch(choice)

{

case 1:

{

add:

printf(" \*\*\* Addition \*\*\*\*");

printf("\nEnter first number::");

scanf("%d", &a);

printf("\nEnter Second number::");

scanf("%d", &b);

result = a+b;

printf("\nAddition is :: %d", result);

printf("\n Do you want to continue ? (0/1)?\n");

scanf("%d",&y);

if (y==1)

{

system("cls");

goto add;

}

else

goto start;

break;

}

case 2:

{

sub:

printf(" \*\*\* Subtraction \*\*\*\*");

printf("\nEnter first number::");

scanf("%d", &a);

printf("\nEnter Second number::");

scanf("%d", &b);

result = a-b;

printf("\nResult is :: %d", result);

printf("\n Do you want to continue ? (0/1)?\n");

scanf("%d",&y);

if (y==1)

{

system("cls");

goto sub;

}

else

goto start;

break;

}

case 3:

{

mul:

printf(" \*\*\* Multiplication \*\*\*\*");

printf("\nEnter first number::");

scanf("%d", &a);

printf("\nEnter Second number::");

scanf("%d", &b);

result = a\*b;

printf("\nResult is :: %d", result);

printf("\n Do you want to continue ? (0/1)?\n");

scanf("%d",&y);

if (y==1)

{

system("cls");

goto mul;

}

else

goto start;

break;

}

case 4:

{

div:

printf(" \*\*\* Division \*\*\*\*");

printf(" \n\*\* Divisor must not be Zero \*\*");

printf("\nEnter first number::");

scanf("%d", &a);

printf("\nEnter Second number::");

scanf("%d", &b);

if (b==0)

{

printf(" Its not possible");

goto start;

}

else

{

result = a/b;

printf("\nQoutient is :: %d", result);

rem = a%b;

printf("\nRemainder is :: %d", rem);

}

printf("\n Do you want to continue ? (0/1)?\n");

scanf("%d",&y);

if (y==1)

{

system("cls");

goto div;

}

else

goto start;

break;

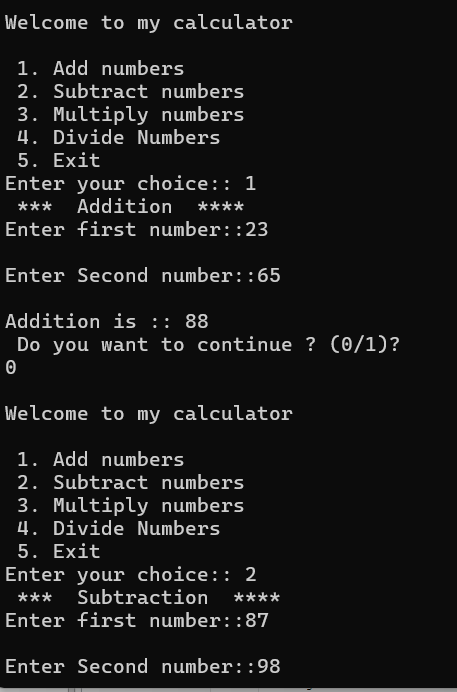
}

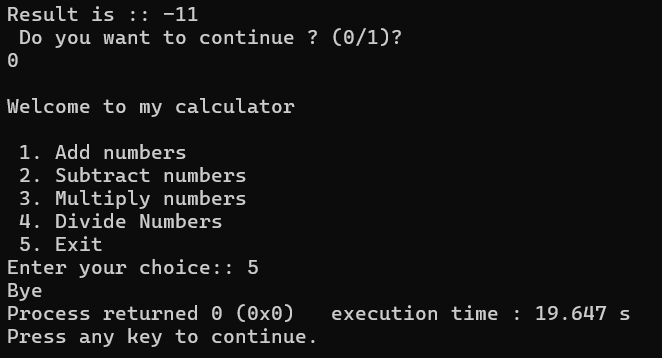
default:

printf("Bye");

}

}





17)write a program to print starts :

#include<stdio.h>

#include<conio.h>

void main()

{

int i,j,ns=0;

system ("cls");

printf("\n Enter the number rows for stars::");

scanf("%d",&ns);

for(i=1;i<=ns;i++)

{

for(j=1;j<=i;j++)

{

printf("\*");

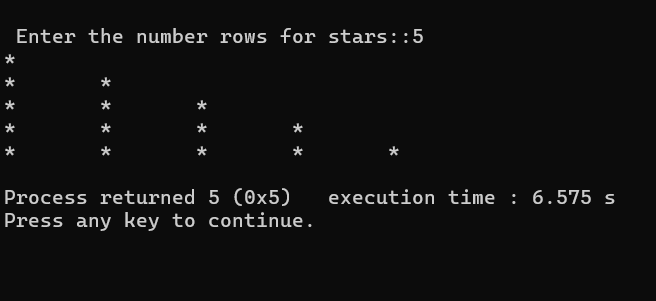
printf("\t");

}

printf("\n");

}

}



18)matrix multiplication

#include<stdio.h>

void main()

{

system ("cls");

int i,j,k,r,a[2][2],b[2][2], c[2][2];

printf("\n Matrix Multiplication [2][2] \n ");

printf("Enter first matrix::");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

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

}

printf("\n");

}

printf("Enter second matrix::");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

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

}

printf("\n");

}

printf("The matrix a[][] and b[][] are :: \n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

printf("%d",a[i][j]);

}

printf("\n");

}

printf("multiplication is :: \n");

for(i=0;i<2;i++)

{

for(j=0;j<2;j++)

{

c[i][j]=0;

for (k=0;k<2;k++)

{

c[i][j] = c[i][j] + ( a[i][k] \* b[k][j]);

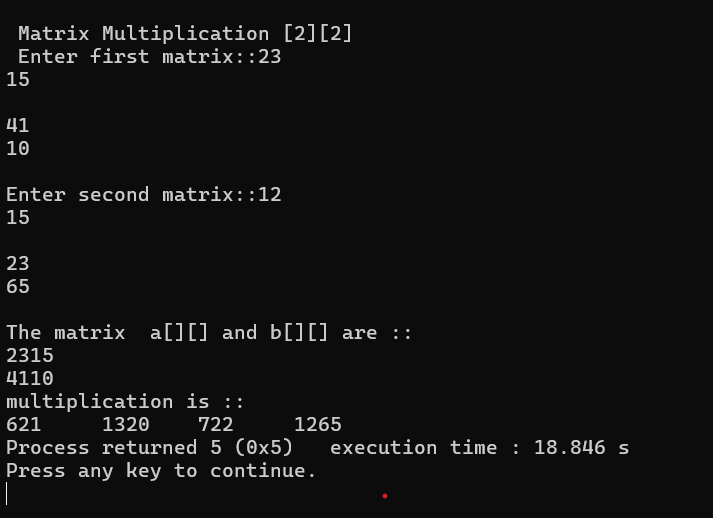
}

printf("%d\t",c[i][j]);

}

}

}



19) trees:

#include <stdio.h>

#include <stdlib.h>

struct Node {

int a;

struct Node \*left;

struct Node \*right;

};

struct Node \*root = NULL;

struct Node\* insert() {

int data;

struct Node \*nn=(struct Node\*)malloc(sizeof(struct Node));

printf("Enter Data [-1 for start inserting left or right]\n");

scanf("%d", &data);

if(data == -1)

return 0;

nn->a = data;

printf("Enter Left Node Data ");

nn->left=insert();

printf("Enter Right Node Data ");

nn->right=insert();

return nn;

}

void preorder(struct Node \*root) {

if (root == NULL) {

return;

}

printf("%d ", root->a);

preorder(root->left);

preorder(root->right);

}

void inorder(struct Node \*root) {

if(root == NULL) {

return ;

}

inorder(root->left);

printf("%d ", root->a);

inorder(root->right);

}

void postorder(struct Node \*root) {

if(root == NULL) {

return ;

}

postorder(root->left);

postorder(root->right);

printf("%d ", root->a);

}

int main() {

root = insert();

printf("Printing the data in the list[preOrder Traversal]\n");

preorder(root);

printf("\nPrinting the data in the list[inOrder Traversal]\n");

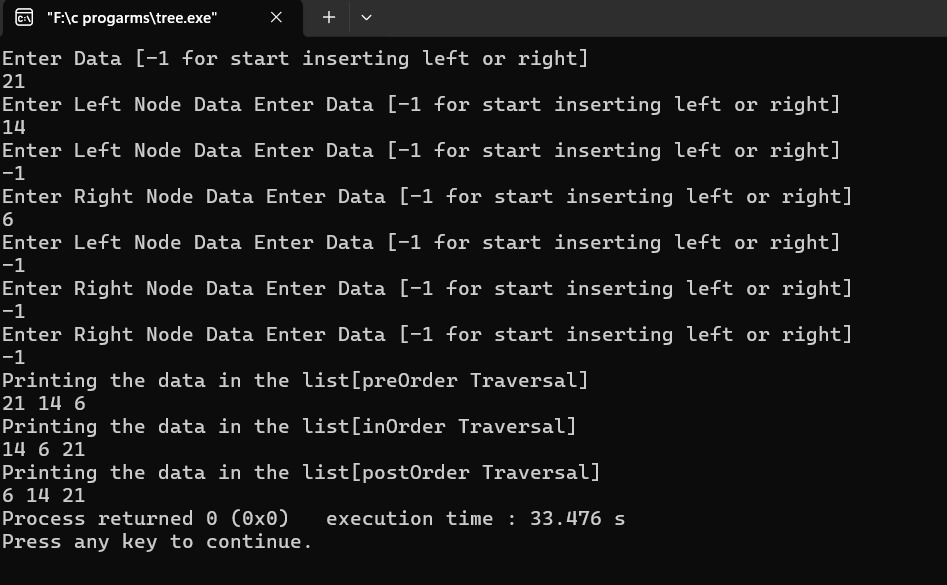
inorder(root);

printf("\nPrinting the data in the list[postOrder Traversal]\n");

postorder(root);

return 0;

}



20) #include<stdio.h>

#include<conio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*left,\*right;

};

typedef struct node btree;

void main()

{

btree \*root= 0;

btree \*create();

root = create();

getch();

}

btree \*create()

{

int x;

btree \*newnode;

newnode=(btree\*)malloc(sizeof(btree));

printf("enter the data OR -1 for no node :");

scanf("%d",&x);

if(x == -1)

return 0;

newnode->data = x;

printf("enter left child of %d\n",x);

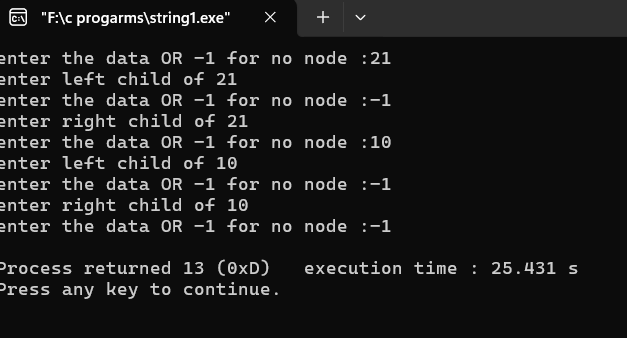
newnode->left = create();

printf("enter right child of %d\n",x);

newnode->right = create();

return newnode;

}



21) struct

#include<stdio.h>

#include<conio.h>

#include<string.h>

struct emp

{

char name[30];

int age;

float salary;

}p1;

main()

{

strcpy (p1.name, "myname");

p1.age = 27;

p1.salary = 10000;

printf("Name:: %s\n", p1.name);

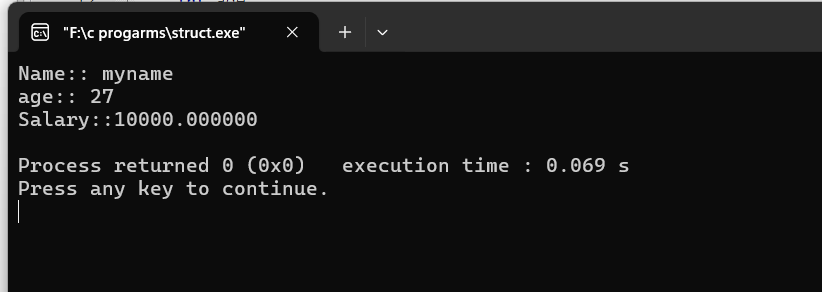
printf("age:: %d\n", p1.age);

printf("Salary::%f\n ", p1.salary);

//printf ("enter your name::");

//scanf

}



22) #include <stdio.h>

int binarySearch(int array[], int x, int low, int high) {

// Repeat until the pointers low and high meet each other

while (low <= high)

{

int mid = low + (high - low) / 2;

if (array[mid] == x)

return mid;

if (array[mid] < x)

low = mid + 1;

else

high = mid - 1;

}

return -1;

}

int main(void) {

int array[] = {3, 4, 5, 6, 7, 8, 9};

int n = sizeof(array) / sizeof(array[0]);

int x = 4;

int result = binarySearch(array, x, 0, n - 1);

if (result == -1)

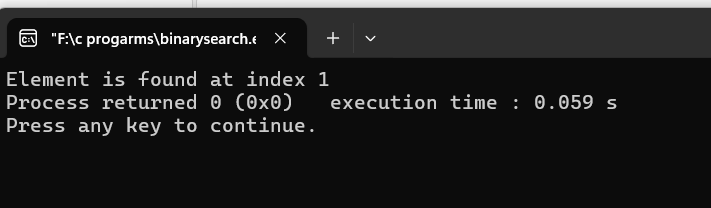
printf("Not found");

else

printf("Element is found at index %d", result);

return 0;

}



24) calendar :

#include <stdio.h>

int dayOfWeek(int d, int m, int y) {

static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};

y -= m < 3;

return (y + y/4 - y/100 + y/400 + t[m-1] + d) % 7;

}

void printCalendar(int month, int year) {

int daysInMonth, i, currentDay;

int days[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))

days[1] = 29;

printf(" \*\*\*\* Calendar - %d/%d \*\*\*\*\n", month, year);

printf(" Sun Mon Tue Wed Thu Fri Sat\n");

currentDay = dayOfWeek(1, month, year);

for (i = 0; i < currentDay; i++)

printf(" ");

for (i = 1; i <= days[month-1]; i++) {

printf("%5d", i);

if (++currentDay > 6) {

currentDay = 0;

printf("\n");

}

}

if (currentDay != 0)

printf("\n");

}

int main() {

int month, year;

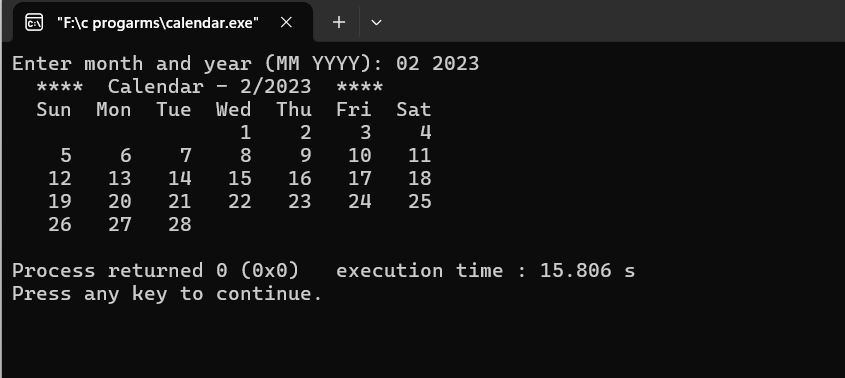
printf("Enter month and year (MM YYYY): ");

scanf("%d %d", &month, &year);

printCalendar(month, year);

return 0;

}



25) call by value

#include <stdio.h>

main()

{

int a,b;

a=5, b=20;

swap (a,b);

swap1 (&a, &b);

printf ("\n Swap Fun: (call by value) \n a = %d , b = %d ", a,b);

printf ("\n Swap1 Fun: (call by Ref) \n a = %d , b = %d ", a,b);

}

void swap (int x, int y)

{

int tmp;

tmp = x;

x=y;

y=tmp;

}

void swap1 (int \*x1, int \*y1)

{

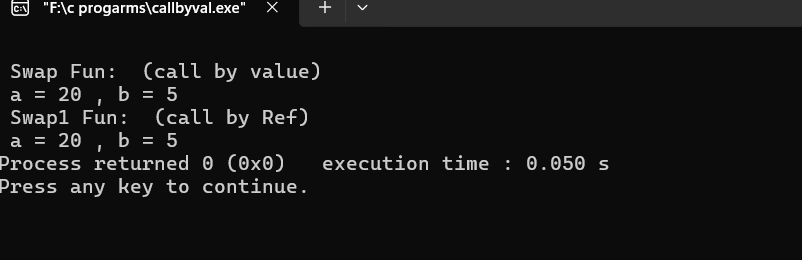
int tmp1;

tmp1 = \*x1;

\*x1=\*y1;

\*y1=tmp1;

}



26) #include<stdio.h>

//#pragma pack(1)

int main(){

struct test

{

char y;

char z;

int x;

};

struct test \*ptr,xyz;

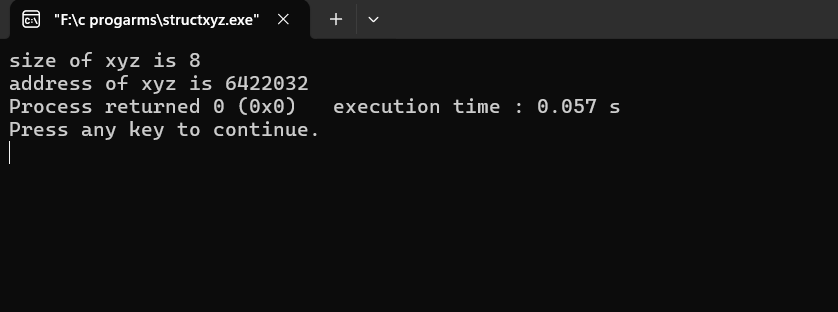
printf("size of xyz is %zu",sizeof(xyz));

ptr=&xyz;

printf("\naddress of xyz is %zu",ptr);

return 0;

}



27) struct1:

#include <stdio.h>

struct person {

int age;

float weight;

};

int main(){

struct person \*personPtr, person1; personPtr = &person1; printf("Enter age: ");

scanf("%d", &personPtr->age);

printf("Enter weight: ");

scanf("%f", &personPtr->weight);

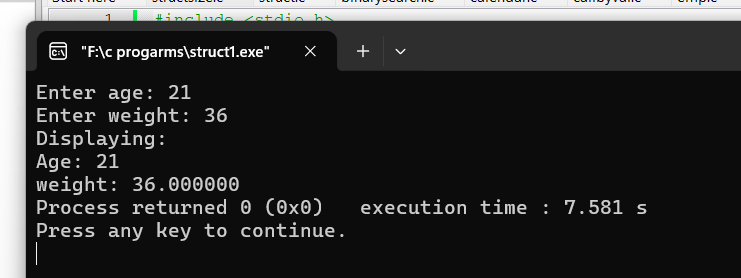
printf("Displaying:\n");

printf("Age: %d\n", personPtr->age);

printf("weight: %f", personPtr->weight);

return 0;

}



28)

#include <stdio.h>

#pragma pack(1)

struct student

{

char name[50];

int age;

};

// function prototype

void display(struct student s);

int main()

{

struct student s1;

printf("the size of student is %zu",sizeof(s1));

printf("\nEnter name: ");

// read string input from the user until \n is entered

// \n is discarded

scanf("%[^\n]%\*c", s1.name);

printf("Enter age: ");

scanf("%d", &s1.age);

display(s1); // passing struct as an argument

return 0;

}

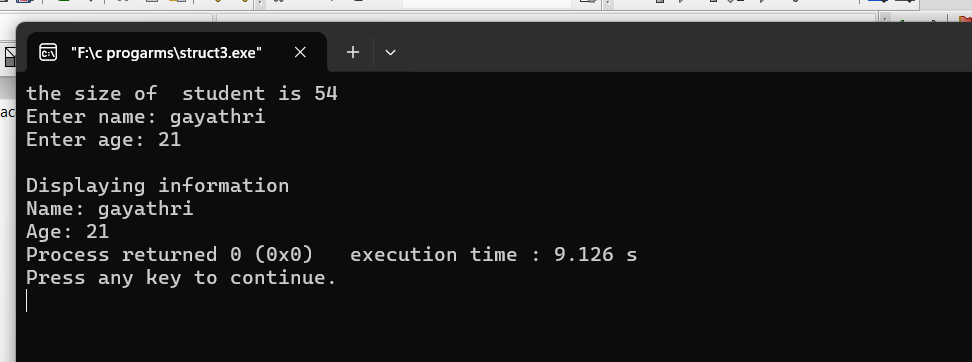
void display(struct student s) {

printf("\nDisplaying information\n");

printf("Name: %s", s.name);

printf("\nAge: %d", s.age);

}



29)// Queue implementation in C

#include <stdio.h>

#define SIZE 5

void enQueue(int);

void deQueue();

void display();

int items[SIZE], front = -1, rear = -1;

int main() {

//deQueue is not possible on empty queue

deQueue();

//enQueue 5 elements

enQueue(1);

enQueue(2);

enQueue(3);

enQueue(4);

enQueue(5);

// 6th element can't be added to because the queue is full

enQueue(6);

display();

//deQueue removes element entered first i.e. 1

deQueue();

//Now we have just 4 elements

display();

return 0;

}

void enQueue(int value) {

if (rear == SIZE - 1)

printf("\nQueue is Full!!");

else {

if (front == -1)

front = 0;

rear++;

items[rear] = value;

printf("\nInserted -> %d", value);

}

}

void deQueue() {

if (front == -1)

printf("\nQueue is Empty!!");

else {

printf("\nDeleted : %d", items[front]);

front++;

if (front > rear)

front = rear = -1;

}

}

// Function to print the queue

void display() {

if (rear == -1)

printf("\nQueue is Empty!!!");

else {

int i;

printf("\nQueue elements are:\n");

for (i = front; i <= rear; i++)

printf("%d ", items[i]);

}

printf("\n");

}int workerNo;

char name[20];

} j;

int main() {

j.salary = 12.3;

// when j.workerNo is assigned a value,// j.salary will no longer hold 12.3 j.workerNo = 100;

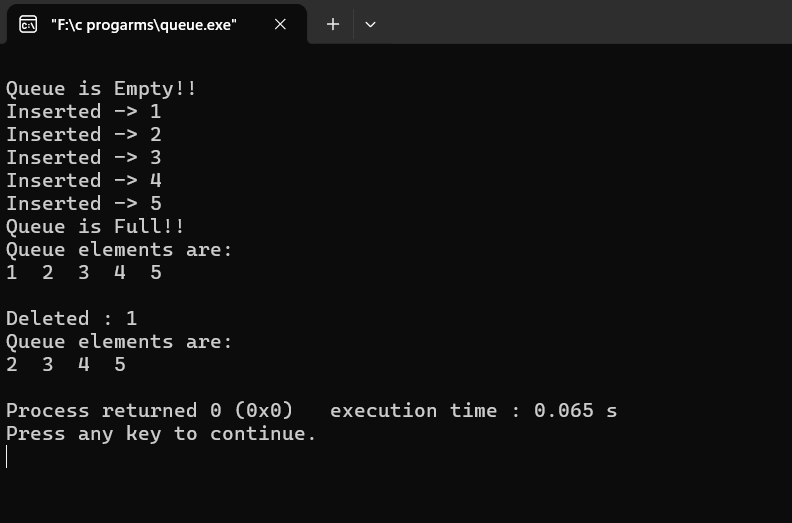
printf("Salary = %.1f\n", j.salary);

printf("Number of workers = %d", j.workerNo);

printf("\nsize of union %zu",sizeof(j));

return 0;

}



30) Arrays:

#include <stdio.h>

void selectionSort(int arr[], int n) {

int i, j, min\_index;

for (i = 0; i < n-1; i++) {

min\_index = i;

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

if (arr[j] < arr[min\_index]) {

min\_index = j;

}

}

int temp = arr[min\_index];

arr[min\_index] = arr[i];

arr[i] = temp;

}

}

void deletionSort(int arr[], int n) {

int i, j;

for (i = 0; i < n-1; i++) {

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

if (arr[j] < arr[i]) {

int k;

for (k = i; k < n-1; k++) {

arr[k] = arr[k+1];

}

n--;

i--;

break;

}

}

}

}

void printArray(int arr[], int n) {

for (int i = 0; i < n; i++) {

printf("%d ", arr[i]);

}

printf("\n");

}

int main() {

int arr[] = {64, 25, 12, 22, 11};

int n = sizeof(arr)/sizeof(arr[0]);

printf("Original array: ");

printArray(arr, n);

selectionSort(arr, n);

printf("Array sorted using Selection Sort: ");

printArray(arr, n);

int arr2[] = {64, 25, 12, 22, 11};

int n2 = sizeof(arr2)/sizeof(arr2[0]);

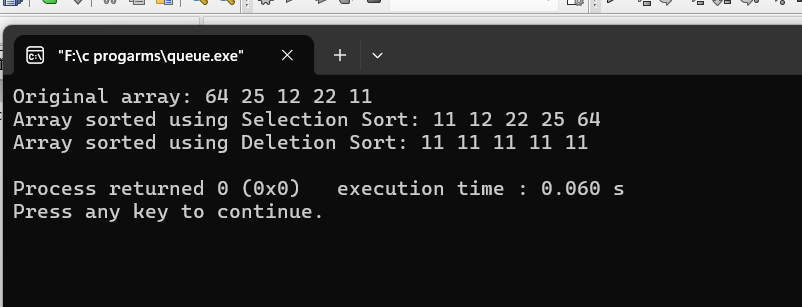
deletionSort(arr2, n2);

printf("Array sorted using Deletion Sort: ");

printArray(arr2, n2);

return 0;

}



31)Current time:

#include <stdio.h>

main()

{

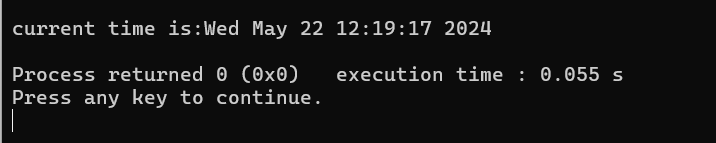
time\_t t;

time(&t);

printf("\ncurrent time is:%\s",ctime(&t));

return 0;

}



32)Linked list insertion from front:

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

struct Node

{

int data;

struct Node\* next;

};

void insertAtFront(struct Node\*\*head\_ref,int new\_data)

{

struct Node\* new\_node=(struct Node\*)malloc(sizeof(struct Node));

new\_node->data = new\_data;

new\_node->next = (\*head\_ref);

(\*head\_ref)=new\_node;

}

void printList(struct Node\* node)

{

while(node!=NULL)

{

printf("%d",node->data);

node=node->next;

}

printf("\n");

}

int main()

{

struct Node\* head=NULL;

insertAtFront(&head,1);

insertAtFront(&head,2);

insertAtFront(&head,3);

insertAtFront(&head,4);

insertAtFront(&head,5);

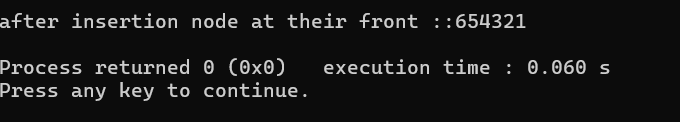
insertAtFront(&head,6);

printf("after insertion node at their front ::");

printList(head);

return 0;

}



33)pointers:

#include <stdio.h>

main()

{

/\*int 1;

float b,c;

double d;

char c;

a=5;b=3.7;c=10.80;d=12345678.99;ch='s';

printf("%d is stored in the location :: %u \n",a,&a);\*/

/\*int \*p,n;

p=&n;

n=0x18;

\*p=\*p+4;

printf("%u is stored in the location :: %u \n",n,&n);

printf("%u is stored in the location :: %u \n",p,&p);

getch();

\*/

/\*&x, &x,

\*ptr, ptr

y, &\*ptr

ptr, &ptr

y, &y\*/

int x,y;

int \*ptr;

x=10;

ptr = &x;

y= \*ptr;

printf ("%d : (x) is stored in location :: %u \n", x, &x);

printf ("%d : (\*&x) is stored in location :: %u \n", \*&x, &x);

printf ("%d : (\*ptr) is stored in location :: %u \n", \*ptr, ptr);

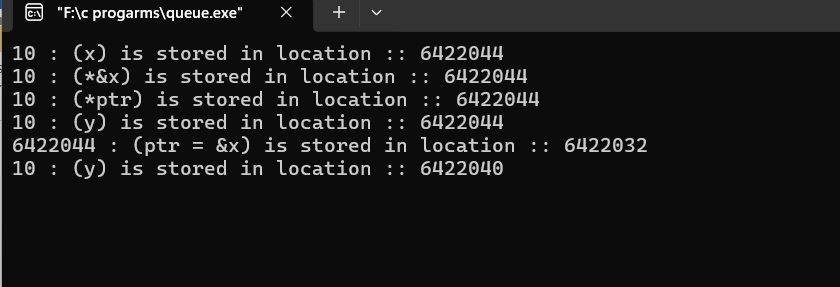
printf ("%d : (y) is stored in location :: %u \n", y, &\*ptr);

printf ("%u : (ptr = &x) is stored in location :: %u \n", ptr, &ptr);

printf ("%d : (y) is stored in location :: %u \n", y, &y);

getch();

}



34)Arguments with no return:

#include<stdio.h>

main()

{

int x,y;

printf("\n enter any two numbers :");

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

sum(x,y); //calling function

getch();

}

void sum(int a ,int b) //called function

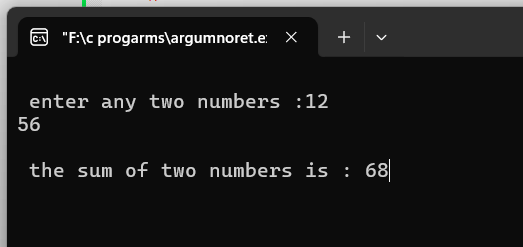
{

int res;

res = a + b;

printf("\n the sum of two numbers is : %d",res);

}



35)function:

#include <stdio.h>

int addNumbers(int a, int b); // function prototype

int main()

{

int n1,n2,sum;

printf("Enters two numbers: ");

scanf("%d %d",&n1,&n2);

sum = addNumbers(n1, n2); // function call

printf("sum = %d",sum);

return 0;

}

int addNumbers(int a, int b) // function definition

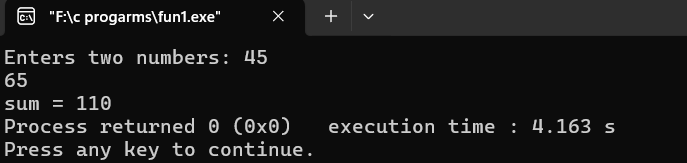
{

int result;

result = a+b;

return result; // return statement

}



36)#include <stdio.h>

void hanoi(int n, char from, char to, char via) {

if(n == 1){

printf("Move disk 1 from %c to %c\n", from, to);

}

else{

hanoi(n-1, from, via, to);

printf("Move disk %d from %c to %c\n", n, from, to);

hanoi(n-1, via, to, from);

}

}

int main() {

int n = 4;

char from = 'A';

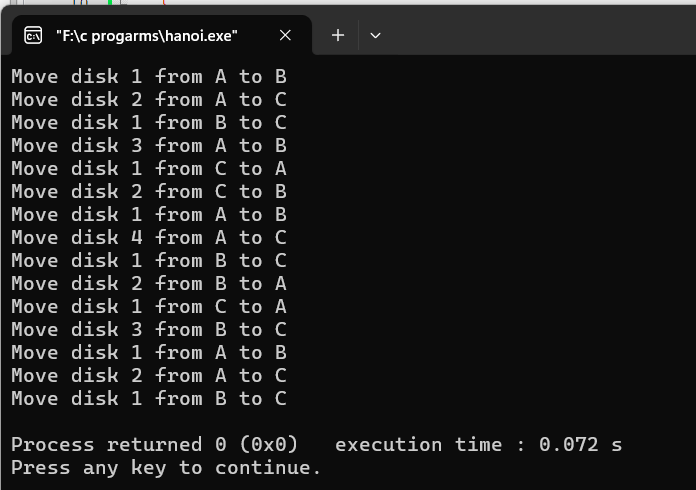
char to = 'B';

char via = 'C';

//calling hanoi() method

hanoi(n, from, via, to);

}



37)file handling:

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

void main ()

{

int n;

FILE \*fptr;

fptr = fopen ("C:\\Files\\output", "a");

if (fptr == NULL)

{

printf ("Error!!!!!");

exit(0);

}

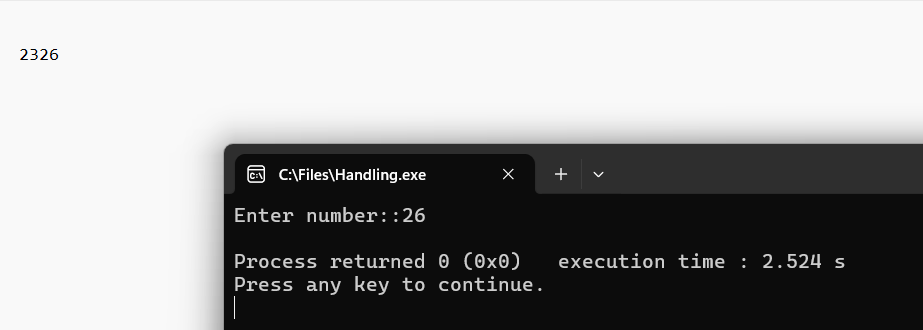
printf ("Enter number::");

scanf ("%d", &n);

fprintf (fptr, "%d", n);

fclose (fptr);

}



38) #include<stdio.h>

#include<conio.h>

#include<string.h>

struct emp

{

char name[10];

int age;

char department[1];

}p1;

main()

{

FILE \*fptr;

fptr = fopen ("C:\\Files\\output", "a");

printf("enter employee name:\n");

scanf("%s",&p1.name);

printf("enter employee age:\n");

scanf("%d",&p1.age);

printf("enter department name:\n");

scanf("%d",&p1.department);

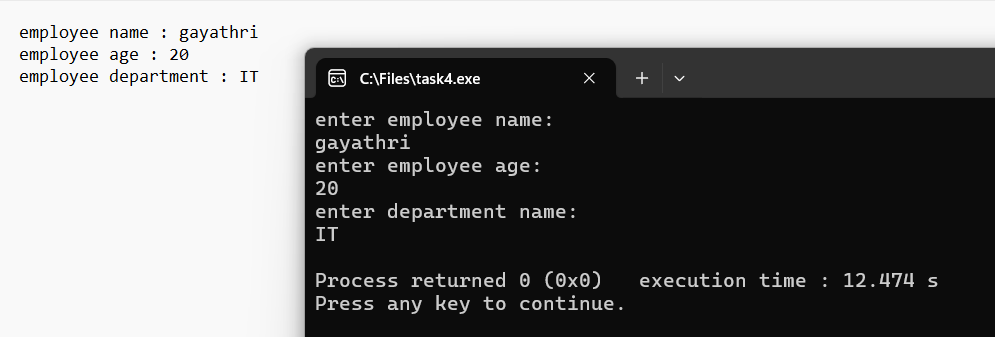
fprintf(fptr,"employee name : %s\n",p1.name);

fprintf(fptr,"employee age : %d\n",p1.age);

fprintf(fptr,"employee department : %s\n",p1.department);

fclose (fptr);

}



39)stack implemention in c:

#include <stdio.h>

#include <stdlib.h>

#define MAX 10

int count = 0;

// Creating a stack

struct stack {

int items[MAX];

int top;

};

typedef struct stack st;

void createEmptyStack(st \*s) {

s->top = -1;

}

// Check if the stack is full

int isfull(st \*s) {

if (s->top == MAX - 1)

return 1;

else

return 0;

}

// Check if the stack is empty

int isempty(st \*s) {

if (s->top == -1)

return 1;

else

return 0;

}

// Add elements into stack

void push(st \*s, int newitem) {

if (isfull(s)) {

printf("STACK FULL");

} else {

s->top++;

s->items[s->top] = newitem;

}

count++;

}

// Remove element from stack

void pop(st \*s) {

if (isempty(s)) {

printf("\n STACK EMPTY \n");

} else {

printf("Item popped= %d", s->items[s->top]);

s->top--;

}

count--;

printf("\n");

}

// Print elements of stack

void printStack(st \*s) {

printf("Stack: ");

for (int i = 0; i < count; i++) {

printf("%d ", s->items[i]);

}

printf("\n");

}

// Driver code

int main() {

int ch,n,ele;

st \*s = (st \*)malloc(sizeof(st));

createEmptyStack(s);

printf(" Enter the size of the stack : ");

scanf("%d",&n);

printf("\n Enter the elements in to the stack : ");

for (int i=0;i<n;i++)

{

scanf("%d",&ele);

push(s, ele);

}

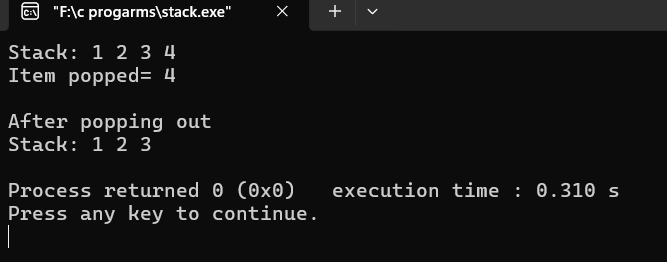
printStack(s);

pop(s);

printf("\nAfter popping out\n");

printStack(s);

}



40)queues:

#include <stdio.h>

main(){

/\*int 1;

float b,c;

double d;

char c;

a=5;b=3.7;c=10.80;d=12345678.99;ch='s';

printf("%d is stored in the location :: %u \n",a,&a);\*/

/\*int \*p,n;

p=&n;

n=0x18;

\*p=\*p+4;

printf("%u is stored in the location :: %u \n",n,&n);

printf("%u is stored in the location :: %u \n",p,&p);

getch();

\*/

/\*&x, &x,

\*ptr, ptr

y, &\*ptr

ptr, &ptr

y, &y\*/

int x,y;

int \*ptr;

x=10;

ptr = &x;

y= \*ptr;

printf ("%d : (x) is stored in location :: %u \n", x, &x);

printf ("%d : (\*&x) is stored in location :: %u \n", \*&x, &x);

printf ("%d : (\*ptr) is stored in location :: %u \n", \*ptr, ptr);

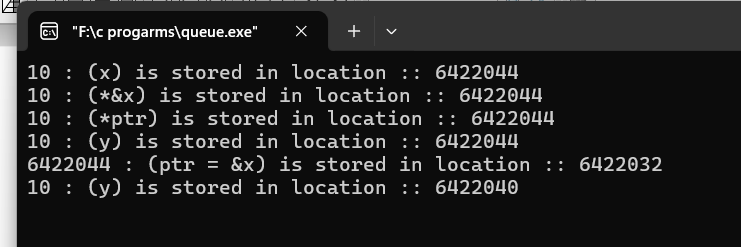
printf ("%d : (y) is stored in location :: %u \n", y, &\*ptr);

printf ("%u : (ptr = &x) is stored in location :: %u \n", ptr, &ptr);

printf ("%d : (y) is stored in location :: %u \n", y, &y);

getch();

}



41)linked list:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Node {

char name[20];

int age;

struct Node \*next;

};

struct Node \*head;

void insert(char name[], int age) {

struct Node \*h = head;

struct Node \*nn = (struct Node\*)malloc(sizeof(struct Node));

nn->age = age;

strcpy(nn->name, name);

nn->next = NULL;

if (h == NULL) {

head = nn;

return;

}

while (h->next != NULL) {

h = h->next;

}

h->next = nn;

}

void display() {

printf("Checking..\n");

struct Node\* h = head;

while (h != NULL) {

printf("\nName: %s and Age: %d\n", h->name, h->age);

h = h->next;

}

}

int main() {

printf("Enter the number of nodes you want to create:\n");

int n;

scanf("%d", &n);

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

int age;

char name[20];

printf("Enter age for node %d\n: ", i);

scanf("%d", &age);

printf("Enter name for node %d\n: ", i);

scanf("%s", name);

insert(name, age);

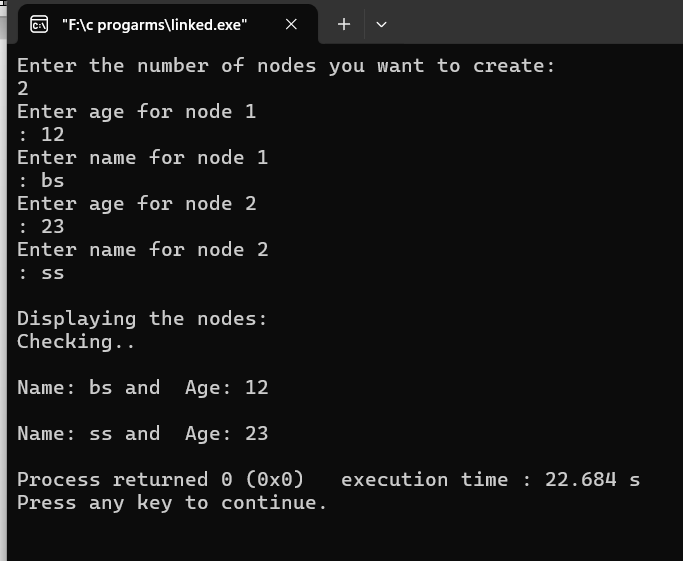
}

printf("\nDisplaying the nodes:\n");

display();

return 0;

}



42) #include <stdio.h>

// Function that returns the index of the

// day for date DD/MM/YYYY

int dayNumber(int day, int month, int year)

{

static int t[] = { 0, 3, 2, 5, 0, 3,

5, 1, 4, 6, 2, 4 };

year -= month < 3;

return (year + year / 4

- year / 100

+ year / 400

+ t[month - 1] + day)

% 7;

}

// Function that returns the name of the

// month for the given month Number

// January - 0, February - 1 and so on

char\* getMonthName(int monthNumber)

{

char\* month;

switch (monthNumber) {

case 0:

month = "January";

break;

case 1:

month = "February";

break;

case 2:

month = "March";

break;

case 3:

month = "April";

break;

case 4:

month = "May";

break;

case 5:

month = "June";

break;

case 6:

month = "July";

break;

case 7:

month = "August";

break;

case 8:

month = "September";

break;

case 9:

month = "October";

break;

case 10:

month = "November";

break;

case 11:

month = "December";

break;

}

return month;

}

// Function to return the number of days

// in a month

int numberOfDays(int monthNumber, int year)

{

// January

if (monthNumber == 0)

return (31);

// February

if (monthNumber == 1) {

// If the year is leap then Feb

// has 29 days

if (year % 400 == 0

|| (year % 4 == 0

&& year % 100 != 0))

return (29);

else

return (28);

}

// March

if (monthNumber == 2)

return (31);

// April

if (monthNumber == 3)

return (30);

// May

if (monthNumber == 4)

return (31);

// June

if (monthNumber == 5)

return (30);

// July

if (monthNumber == 6)

return (31);

// August

if (monthNumber == 7)

return (31);

// September

if (monthNumber == 8)

return (30);

// October

if (monthNumber == 9)

return (31);

// November

if (monthNumber == 10)

return (30);

// December

if (monthNumber == 11)

return (31);

}

// Function to print the calendar of

// the given year

void printCalendar(int year)

{

printf(" Calendar - %d\n\n", year);

int days;

// Index of the day from 0 to 6

int current = dayNumber(1, 1, year);

// i for Iterate through months

// j for Iterate through days

// of the month - i

for (int i = 0; i < 12; i++) {

days = numberOfDays(i, year);

// Print the current month name

printf("\n ------------%s-------------\n",

getMonthName(i));

// Print the columns

printf(" Sun Mon Tue Wed Thu Fri Sat\n");

// Print appropriate spaces

int k;

for (k = 0; k < current; k++)

printf(" ");

for (int j = 1; j <= days; j++) {

printf("%5d", j);

if (++k > 6) {

k = 0;

printf("\n");

}

}

if (k)

printf("\n");

current = k;

}

return;

}

// Driver Code

int main()

{

int year = 2016;

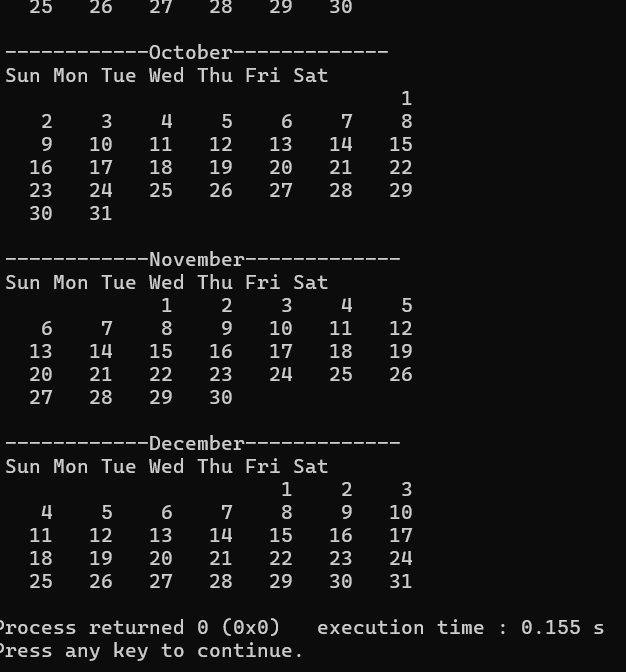
// Function Call

printCalendar(year);

return 0;

}





43)while:

#include<stdio.h>

#include<conio.h>

void main(){

int a=5;

while(a!=0) {

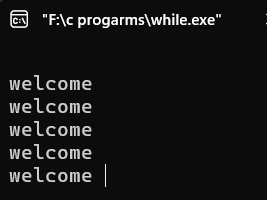
printf("\n welcome ");

a--;

}

getch();

}



44)two dimensional array:

#include <stdio.h>

int main()

{

char usernames[21][20] = {

"siri", "divya", "malleswari", "madhavi", "nithya",

"keerthi", "fazilath", "mounika", "lakshmi", "gayathri",

"madhuri", "lohitha", "neeraja", "suma", "parimala",

"bhavani", "pallavi", "bhuvana", "likitha", "kiranmai",

"lalitha"

};

// Print the user names

for (int i = 0; i < 21; i++)

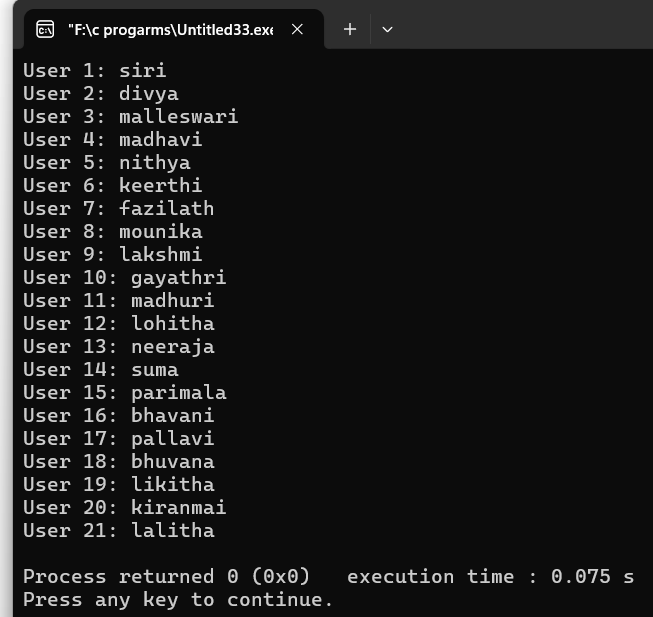
{

printf("User %d: %s\n", i + 1, usernames[i]);

}

return 0;

}



45) #include <stdio.h>

#include <conio.h>

main()

{

char name[30];

printf("\nEnter Your name ::");

//scanf ("%s", name);

fgets(name, sizeof(name), stdin);

printf("\nYour name is :: %s", name);

printf("\nwith puts function:: ");

puts(name);

getch();

}

