|  |  |  |
| --- | --- | --- |
| **Name: Naren Chandrashekhar** | **SRN: PES2UG20CS216** | **Section: G** |
| **Date: 10/06/2021** | **Week Number: 6** |

|  |  |
| --- | --- |
| **1** | **1) Write a C program to generate Pascal triangle using two dimensional array**  **Input:**  Enter the n value:  4  **Output:**  1  1 1  1 2 1  1 3 3 1 |
|  | **Program:**  **#include<stdio.h>**  **#include<conio.h>**  **void pascaltriangle(int n; int a[][n], int n);**  **void displaypascal(int n; int a[][n], int n);**  **int main()**  **{**  **int a[50][50],n;**  **printf("Enter the value of n ");**  **scanf("%d",&n);**  **pascaltriangle(a,n);**    **return 0;**  **}**  **void pascaltriangle(int n; int a[][n], int n)**  **{**  **int i,j;**  **for(i=1;i<=n;i++)**  **{**  **for(j=1;j<=i;j++)**  **{**  **if(j==1 || j==i)**  **{**  **a[i][j] = 1;**  **}**  **else**  **{**  **a[i][j] = a[i-1][j]+a[i-1][j-1];**  **}**  **}**  **}**  **displaypascal(a,n);**  **}**  **void displaypascal(int n; int a[][n], int n)**  **{**  **int i,j,k;**  **for(i=1;i<=n;i++)**  **{**  **for(j=1;j<=i;j++)**  **{**  **printf("%d ",a[i][j]);**  **}**  **printf("\n");**  **}**  **}** |
|  | **Output Screenshot:** |
| **2** | Write a C program to read elements in a matrix and check whether the given matrix is symmetric matrix or not.  **Input:**  Enter the value of m  3  Enter the value of n  3  Enter elements in matrix of size 3x3:  1  0  0  0  1  0  0  0  1  **Output:**  The given matrix is Symmetric matrix:  1 0 0  0 1 0  0 0 1 |
|  | **Program:**  **#include<stdio.h>**  **#include<conio.h>**  **int main()**  **{**  **int a[50][50],n,m;**  **printf("Enter the value of m ");**  **scanf("%d",&m);**  **printf("Enter the value of n ");**  **scanf("%d",&n);**  **printf("Enter the elements of the %dx%d matrix ",m,n);**  **int i,j;**  **for(i=1;i<=m;i++)**  **{**  **for(j=1;j<=n;j++)**  **{**  **scanf("%d",&a[i][j]);**  **}**  **}**  **int r,c,b[50][50];**  **for(r=1;r<=m;r++)**  **{**  **for(c=1;c<=n;c++)**  **{**  **b[c][r] = a[r][c];**  **}**  **}**  **int isSymmetric = 1;**  **for(i=1,r=1;i<=m,r<=m;i++,r++)**  **{**  **for(j=1,c=1;j<=n,c<=n;j++,c++)**  **{**  **if(a[i][j] == b[r][c])**  **continue;**  **else**  **{**  **isSymmetric = 0;**  **break;**  **}**  **}**  **}**  **if(isSymmetric)**  **printf("The matrix is symmetric ");**  **else**  **printf("The matrix is not symmetric ");**  **return 0;**  **}** |
|  | **Output Screenshot:** |
| **3** | Write a C program to compare 2 dates and print appropriate message using structures  **Input1:**  Enter Date1 in the format dd/mm/yyyy  12/2/2000  Enter Date2 in the format dd/mm/yyyy  12/2/2000  Date1=12/2/2000  Date2=12/2/2000  **Output1:**  Date1 is equal to Date2  Input2:  Enter Date1 in the format dd/mm/yyyy  12/3/2000  Enter Date2 in the format dd/mm/yyyy  12/3/2001  Date1=12/3/2000  Date2=12/3/2001  **Output2:**  Date1 is smaller than Date2  **Input3:**  Enter Date1 in the format dd/mm/yyyy  12/4/1999  Enter Date2 in the format dd/mm/yyyy  12/2/1999  Date1=12/4/1999  Date2=12/2/1999  **Output3:**  Date1 is greater than Date2 |
|  | **Program:**  **#include<stdio.h>**  **#include<conio.h>**  **typedef struct date**  **{**  **int dd;**  **int mm;**  **int yyyy;**  **}date\_info;**  **void date\_read(date\_info \*d);**  **void display\_date(date\_info \*d);**  **int date\_cmp(const date\_info \*d1, const date\_info \*d2);**  **int main()**  **{**  **date\_info d1,d2;**    **printf("Enter Date1 ");**  **date\_read(&d1);**  **printf("Enter Date2 ");**  **date\_read(&d2);**  **printf("\nDate1 = ");**  **display\_date(&d1);**  **printf("\nDate2 = ");**  **display\_date(&d2);**    **int res = date\_cmp(&d1,&d2);**  **if(res==0)**  **printf("\nDate1 is equal to Date2");**  **else if(res>0)**  **printf("\nDate1 is greater than Date2");**  **else**  **printf("\nDate1 is less than Date2");**  **return 0;**  **}**  **void date\_read(date\_info \*d)**  **{**  **scanf("%d",&d->dd);**  **scanf("%d",&d->mm);**  **scanf("%d",&d->yyyy);**  **}**  **void display\_date(date\_info \*d)**  **{**  **printf("%d/%d/%d\n",d->dd, d->mm, d->yyyy);**  **}**  **int date\_cmp(const date\_info \*d1, const date\_info \*d2)**  **{**  **int res;**  **if(d1->dd == d2->dd && d1->mm == d2->mm && d1->yyyy == d2->yyyy)**  **res = 0;**  **else if(d1->yyyy > d2->yyyy || d1->mm > d2->mm && d1->yyyy == d2->yyyy || d1->dd > d2->dd && d1->mm == d2->mm && d1->yyyy == d2->yyyy)**  **res = 1;**  **else**  **res = -1;**  **return res;**  **}** |
|  | **Output Screenshot:** |
| 4 | Write a C Program to Add and subtract two Complex Numbers by Passing Structure to a Function  **Input:**  For 1st complex number  Enter the real and imaginary parts: 5  4  For 2nd complex number  Enter the real and imaginary parts: 3  2  **Output:**  Sum = 8.0 + 6.0i  Sub = 2.0 - 2.0i |
|  | Program:  #include<stdio.h>  #include<conio.h>  typedef struct complex  {  int real;  int img;  }complex\_num;  void read\_values(complex\_num \*c);  void display\_complex(complex\_num \*c);  void add(complex\_num \*c1,complex\_num \*c2);  void sub(complex\_num \*c1,complex\_num \*c2);  int main()  {  complex\_num c1,c2;  printf("Enter the real and imaginary part for first complex number: ");  read\_values(&c1);  printf("Enter the real and imaginary part for second complex number: ");  read\_values(&c2);  display\_complex(&c1);  display\_complex(&c2);  add(&c1,&c2);  sub(&c1,&c2);  return 0;  }  void read\_values(complex\_num \*c)  {  scanf("%d %d",&c->real,&c->img);  }  void display\_complex(complex\_num \*c)  {  printf("%d + %di\n",c->real,c->img);  }  void add(complex\_num \*c1,complex\_num \*c2)  {  complex\_num sum;  sum.real = c1->real + c2->real;  sum.img = c1->img + c2->img;  printf("The sum is %d + %di\n",sum.real,sum.img);  }  void sub(complex\_num \*c1,complex\_num \*c2)  {  complex\_num sub;  sub.real = c1->real - c2->real;  sub.img = c1->img - c2->img;  if(sub.img>0)  printf("The difference is %d + %di\n",sub.real,sub.img);  else  printf("The difference is %d %di\n",sub.real,sub.img);  } |
|  | Output Screenshot: |
| 1 | **Practice Programs**  Write a program that fills a five-by-five matrix as follows:  Upper left triangle with +1s  Lower right triangle with -1s  Right to left diagonal with zeros  Display the contents of the matrix using not more than two printf statements  **Output:**  This is 5x5 Matrix  1 1 1 1 0  1 1 1 0 -1  1 1 0 -1 -1  1 0 -1 -1 -1  0 -1 -1 -1 -1 |
|  | Program: |
|  | Output Screenshot: |
| 2 | Write a Program to add two distances in the inch-feet system using structures  **Input:**  Enter 1st distance  Enter feet: 23  Enter inch: 10  Enter 2nd distance  Enter feet: 34  Enter inch: 2.4  **Output:**  Sum of distances = 58'-0.4" |
|  | Program: |
|  | Output Screenshot: |