|  |  |  |
| --- | --- | --- |
| **Name:**  **P K Navin Shrinivas** | **SRN:PES2UG20CS237** | **Section:G2(B-1)** |
| **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>  int main(){  int n;  printf("Enter value of n : ");  scanf("%i",&n);  int a[n][n];  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  {  if(j==0)  \*(\*(a+i)+j)=1;  else if(j>i)  \*(\*(a+i)+j)=0;  else if(j>=i)  \*(\*(a+i)+j)=1;  else  \*(\*(a+i)+j)=i;  }  }  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  {  if(\*(\*(a+i)+j)!=0)  printf("%i",\*(\*(a+i)+j));  }  printf("\n");  }  } |
|  | **Output Screenshot:**  **1** |
| **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>  int main(){  int n,m;  printf("Enter value of m : ");  scanf("%i",&m);  printf("Enter value of n : ");  scanf("%i",&n);  int a[m][n];  printf("Enter values of the array [%ix%i] from left to right top to bottom : \n",m,n);  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  scanf("%i",\*(a+i)+j);  }  int flag=1;  if(n!=m)  flag=0;  else  {  for(int i=0;i<n;i++)  for(int j=0;j<n;j++)  {  if(\*(\*(a+i)+j)!=\*(\*(a+j)+i))  flag=0;  }  }  }  if(flag)  printf("The given matrix is symmetric : \n");  else  printf("Given matrix is not symmetric : \n");  for(int i=0;i<n;i++)  {  for(int j=0;j<n;j++)  {  printf("%i",\*(\*(a+i)+j));  printf(" ");  }  printf("\n");  }  } |
|  | **Output Screenshot:**  **2_1** |
|  | **2_2** |
| **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<stdlib.h>  int main()  {  char d1[12],d2[12];  *//2 extra for delimeters and new line cus of fgets , gcc compiler in linux shouts that using gets()*  *//is dangerous*  printf("Enter date 1 in the format dd/mm/yyyy : ");  fgets(d1,12,stdin);  printf("Enter date 2 in the format dd/mm/yyyy : ");  fgets(d2,12,stdin);d1[10]='\0';d2[10]='\0';*//line feed gives extra newlines in puts, less remove them*  printf("DATE 1 : ");  puts(d1);  printf("DATE 2 : ");  puts(d2);  char years1[5]={d1[6],d1[7],d1[8],d1[9],'\0'};  int year1=atoi(years1);  char years2[5]={d2[6],d2[7],d2[8],d2[9],'\0'};  int year2=atoi(years2);  char months1[3]={d1[3],d1[4],'\0'};  int month1=atoi(months1);  char months2[3]={d2[3],d2[4],'\0'};  int month2=atoi(months2);  char days1[3]={d1[0],d1[1],'\0'};  int day1=atoi(days1);  char days2[3]={d2[0],d2[1],'\0'};  int day2=atoi(days2);  if(year1==year2 && month1==month2 && day1==day2)  {  printf("The dates are equal!");  return 0;  }  else if(year1>year2)  {  printf("Date 1 is greater than Date 2");  return 0;  }  else if(month1>month2)  {  printf("Date 1 is greater than Date 2");  return 0;  }  else if(day1>day2)  {  printf("Date 1 is greater than Date 2");  return 0;  }  printf("Date 2 is greater than Date 1");  return 0;  } |
|  | **Output Screenshot:**  **3** |
| 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>  struct complex{  int real;  int imaginary;  };  void comsum(struct complex *complex1*,struct complex *complex2*)  {  if(complex1.imaginary+complex2.imaginary>0)  printf("Sum :%i+%ii \n",*complex1*.real+*complex2*.real,*complex1*.imaginary+*complex2*.imaginary);  else  printf("Sum :%i%ii \n",*complex1*.real+*complex2*.real,*complex1*.imaginary+*complex2*.imaginary);  }  void comsub(struct complex *complex1*,struct complex *complex2*)  {  if(complex1.imaginary-complex2.imaginary>0)  printf("Sub :%i+%ii",*complex1*.real-*complex2*.real,*complex1*.imaginary-*complex2*.imaginary);  else  printf("Sub :%i%ii",*complex1*.real-*complex2*.real,*complex1*.imaginary-*complex2*.imaginary);  }  int main(){  struct complex com1,com2;  printf("Enter real and imaginary part of first complex 1: ");  scanf("%i",&(*com1*.real));  scanf("%i",&(*com1*.imaginary));  printf("Enter real and imaginary part of complex 2: ");  scanf("%i",&(*com2*.real));  scanf("%i",&(*com2*.imaginary));  comsum(com1,com2);  comsub(com1,com2);  } |
|  | Output Screenshot:  4 |
| 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:  #include<stdio.h>  int main(){  int a[5][5];  for(int i=0;i<5;i++)  {  for(int j=0;j<5;j++)  {  if(i==5-j)  \*(\*(a+i)+j)=0;  else if(i<5-j)  \*(\*(a+i)+j)=1;  else  \*(\*(a+i)+j)=-1;  }  }  for(int i=0;i<5;i++)  {  for(int j=0;j<5;j++)  {  if(\*(\*(a+i)+j)==-1)  printf("%i",\*(\*(a+i)+j));  else  printf(" %i",\*(\*(a+i)+j));  }  printf("\n");  }  } |
|  | Output Screenshot:  5 |
| 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:  #include<stdio.h>  void fixinches(double \**foot*,double\* *inch*)  {  if(\*inch>=12)  {  int newfoot=(\*inch)/12;*//i like this solution taking advantage of type conversions!*  \*inch=(\*inch)-(newfoot\*12);  \*foot+=newfoot;  }  }  int main()  {  double foot1,inch1,foot2,inch2;  printf("Enter 1st distance :\n");  printf("Foot :");  scanf("%lf",&foot1);  printf("Inch :");  scanf("%lf",&inch1);  printf("Enter 2nd distance :\n");  printf("Foot :");  scanf("%lf",&foot2);  printf("Inch :");  scanf("%lf",&inch2);  fixinches(&foot1,&inch1);  fixinches(&foot2,&inch2);  double tfoot=foot1+foot2;  double tinch=inch1+inch2;  fixinches(&tfoot,&tinch);  printf("Sum of distance : %.2lf\' %.2lf\"",tfoot,tinch);  } |
|  | Output Screenshot:  6 |