|  |  |
| --- | --- |
| **Name** | **PRATIK PUJARI** |
| **UID no.** | **2020300054** |
| **Experiment No.** | 8 |

|  |  |
| --- | --- |
| **AIM:** | To apply the concepts of structure/union to solve a given problem. |
| **Program 1** | |
| **PROBLEM STATEMENT :** | A men’s sports club keeps elaborate computerized records of all its members. The records contain typical information such as Name, age, address, etc. of each person. But there is also information about whether a member is an active playing members, about whether he is married, and so on.  If he is married the record contains information about his wife’s name, the no. of children and their names. Write a program which demonstrates how such a system might be implemented. Show how the names of the wives of all active playing members might be printed. |
| **PROGRAM:** | ALGORITHM:  void main():  STEP 1: START.  STEP 2: Initialize the loop counter i,mem and input the number of members and store it in the variable to mem.  STEP 3: Initialize a struct of array person[mem] in club structure.  STEP 4: Call the predefined function get\_info(person,mem) and display(person,mem).  STEP 5: END.    void get\_info(struct club person[],int n)  STEP 1: START.  STEP 2: Initialize the loop counter i and variable marr\_status.  STEP 3: For i=0 and less than n Repeat steps 3.1,3.2,3.3,3.4,3.5,3.6 and 3.7 or else go to step 4 if the condition fails.  STEP 3.1: Printf(“Enter the name of the member.”) and input the string from the user using gets and store it in person[i].name.  STEP 3.2: Printf(“Enter the age of the member :”) and input the string from the user using gets and store it in person[i].name.  STEP 3.3: Printf(“Enter the address of the member”) and input the string from the user and store it in person[i].address.  STEP 3.4: Printf(“Enter “active ”if the player is active and “inactive” if the player is inactive”) and input the string from the user and store it in person[i].status.  STEP 3.5: Printf("If the person is marraried press '1' else press '2':") and input the value from the user and store it in marr\_status.  STEP 3.6: If marr\_status is equal to 1 the call the predefined function get\_marriage\_info(person,i) or else go to next step.  STEP 3.7: Increment the loop counter by one (i).  STEP 4: END.  void get\_marriage\_info(struct club person[],int i)  STEP 1: START.  STEP 2: Initialize the variables nchildren and loop counter j.  STEP 3: Printf("Enter the wife's name:") and input the string from user using gets(person[i].wife\_name).  STEP 4: Printf("Enter the number of children:") and input the value from user and store it in person[i].children.  STEP 5: Do nchildren=person[i].children.  STEP 6: For j=0 and less the nchildren ,Repeat the steps 6.1,6.2 and 6.3  or else if the condition fails go to step 7.  STEP 6.1: Printf("Enter the name of child%d:",j+1).  STEP 6.2: scanf("%[^\n]%\*c",&person[i].child\_name[i])  STEP 6.3: Increment the loop counter j by one.  STEP 7: END.  void display\_mem(struct club person[],int n)  PROGRAM:  #include<stdio.h>  #include<string.h>  struct club  {  char name[20];  int age;  char address[50];  char status[10];  char wife\_name[20];  int children;  char child\_name[][20];  };  void get\_marriage\_info(struct club person[],int n);  void get\_info(struct club person[],int n)  {  int i,marr\_status;  for(i=0;i<n;i++)  {  fflush(stdin);  printf("Enter the name of Member%d:",i+1);  gets(person[i].name);  printf("Enter the Age of the Member%d:",i+1);  scanf("%d",&person[i].age);  printf("Enter the Address of the Member%d:",i+1);  fflush(stdin);  gets(person[i].address);  printf("Enter 'active' if the player is active or 'inactive' if the player is inactive:");  scanf("%s",person[i].status);  printf("If the person is marraried press '1' else press '2':");  scanf("%d",&marr\_status);  if(marr\_status==1)  {  get\_marriage\_info(person,i);  }  }  }  void get\_marriage\_info(struct club person[],int i)  {  int nchildren,j;  fflush(stdin);  printf("Enter the wife's name:");  gets(person[i].wife\_name);  fflush(stdin);  printf("Enter the number of children:");  scanf("%d",&person[i].children);  nchildren=person[i].children;  for(j=0;i<nchildren;i++)  {  fflush(stdin);  printf("Enter the name of child%d:",i+1);  scanf("%[^\n]%\*c",&person[i].child\_name[i]);  }  }  void display\_mem(struct club person[],int n)  {  int i,j;  printf("-------------- Member of Men Sports Club-------------- \n");  for(i=0;i<n;i++)  {  if(strcmp(person[i].status,"active")==0)  {  printf("Name of Member%d:",i+1);  printf("%s\n",person[i].name);  printf("Name of Member's Wife name%d:",i+1);  printf("%s",person[i].wife\_name);  }  else  printf("No active players found");  }  }  void main()  {  int i,mem;  printf("Enter the number of members in the club:");  scanf("%d",&mem);  struct club person[mem];  get\_info(person,mem);  display\_mem(person,mem);  } |
| **RESULT:** | |
| **INPUT:** |  |
| **OUTPUT:** |  |

|  |  |
| --- | --- |
| **Program 2** | |
| **PROBLEM STATEMENT :** | An airline reservation system maintains records for possible flights consisting of  STARTING POINT 3 character code  DESTINATION 3 character code  STARTING TIME integer on scale 0001 – 2400  ARRIVAL TIME integer on scale 0001 – 2400  SEATS positive integer in suitable range.  Your program is to read 20 such records followed by queries of the form STARTING  POINT– DESTINATION, one to a line. For each query find whether there is a possible flight with a seat available; if so reduce the number of seats by one and print out the flight details (or an apology). |
| **PROGRAM:** | ALGORITHM:  void main():  STEP 1: START  STEP 2: Initialize the variable num\_flight .  STEP 3: Printf("Enter the number of flights to be recorded:") and store it in the variable num\_flights.  STEP 4: Call the predefined function flight\_details(list,num\_flight) and Customer\_input(list,num\_flight).  STEP 5: END.  void flight\_details(struct flight\_info list[] ,int num\_flight)  STEP 1: START.  STEP 2: Initialize the loop counter i to zero.  STEP 3: For i=0 and less than num\_flight Repeat the steps 3.1,3.2,3.3,3.4,3.5 ,3.6 and 3.7 or else if the condition fails go to step 4.  STEP 3.1: Printf("Entering the details for Flight no.%d",i+1).  STEP 3.2: Printf("\nEnter the STARTING POINT of the flight:") and input the string from the user and store it in list[i].start\_pt.  STEP 3.3: Printf("Enter the DESTINATION POINT of the flight:") and input the string from the user and store it in list[i].des\_pt.  STEP 3.4: Printf("Enter the STARTING TIME of the flight =") and input the string from the user and store it in list[i].start\_time.  STEP 3.5: Printf("Enter the ARRIVAL TIME of the flight =") and input the string from the user ands store it in list[i].arr\_time.  STEP 3.6: Printf("Enter the number of the seats available in the flight:") and input the string from the user and store it in list[i].no\_seats.  STEP 4: Printf(“Details saved..”).  STEP 5: END.  void Customer\_input(struct flight\_info list[],int num\_flight)  STEP 1: START.  STEP 2: Declare the char array strt\_pt[4],des\_pt[4],last\_check[4] and integer variable seats.  STEP 3: Printf("\nEnter the STARTING POINT of the flight:") and input the string from the user and store it in strt\_pt char array.  STEP 4: Printf(“Enter the DESTINATION POINT of the flight:”) and input the string from the user and store it in des\_pt char array.  STEP 5: Initialize the variable check\_pt = flight\_checker(list,strt\_pt,des\_pt,num\_flight).  STEP 6: If check\_pt greater than zero then execute 6.1,6.2,6.3,6.4 and 6.5 or else go to step 7.  STEP 6.1: Printf("\nThe Timings for the flights is:\n(Starting Time: %s Arrival Time: %s)",list[check\_pt].start\_time,list[check\_pt].arr\_time)  STEP 6.2: If (list[check\_pt].no\_seats>0) then execute 6.2.1 and 6.2.2 or else go to step 6.2.3.  STEP 6.2.1: Printf("\nThe seats available for this flight is:%d",list[check\_pt].no\_seats).  STEP 6.2.2: Printf("\nEnter the number of seats to be reserved:") and store the user input to seats and do list[check\_pt].no\_seats-=seats  STEP 6.2.3: Printf("\nNo seats available.... Please try again later.").  STEP 6.3: Printf("\nDo you want to make a another reservation press yes or no:") and input the user string and store it in last\_check  STEP 6.4: If strcmp(last\_check,"yes") equal to zero then call the predefined function Customer\_input(list,num\_flight) or else go to step 6.5.  STEP 6.5: Printf("\nTravel safe and hope that you have a great journey")  STEP 7: Printf(“No Flights Found…”).  STEP 8: END  PROGRAM:  #include<stdio.h>  #include<string.h>  struct flight\_info  {  char start\_pt[4],des\_pt[4];  char start\_time[12],arr\_time[12];  int no\_seats;  }list[20];  int flight\_checker(struct flight\_info list[],char start[], char end[],int nflight)  {  int i,flag;  for(i=0;i<nflight;i++)  {  if(strcmp(list[i].start\_pt,start)==0)  {  flag=i;  break;  }  else  return -1;  }  return flag;  }  void flight\_details(struct flight\_info list[] ,int num\_flight)  {  int i;  for(i=0;i<num\_flight;i++)  {  printf("----------------------Entering the details for Flight no.%d----------------------",i+1);  printf("\nEnter the STARTING POINT of the flight:");  scanf("%s",list[i].start\_pt);  printf("Enter the DESTINATION POINT of the flight:");  scanf("%s",list[i].des\_pt);  fflush(stdin);  printf("Enter the STARTING TIME of the flight =");  gets(list[i].start\_time);  fflush(stdin);  printf("Enter the ARRIVAL TIME of the flight =");  gets(list[i].arr\_time);  printf("Enter the number of the seats available in the flight:");  scanf("%d",&list[i].no\_seats);  }  printf("Details Saved....");  }  void Customer\_input(struct flight\_info list[],int num\_flight)  {  char strt\_pt[4],des\_pt[4],last\_check[4];  int seats;  printf("\n----------------------BOOKING MODE----------------------");  printf("\nEnter the STARTING POINT of the flight:");  scanf("%s",strt\_pt);  printf("Enter the DESTINATION POINT of the flight:");  scanf("%s",des\_pt);  int check\_pt=flight\_checker(list,strt\_pt,des\_pt,num\_flight);  if(check\_pt>=0)  {  printf("Flights found....");  printf("\nThe Timings for the flights is:\n(Starting Time: %s Arrival Time: %s)",list[check\_pt].start\_time,list[check\_pt].arr\_time);  if(list[check\_pt].no\_seats>0)  {  printf("\nThe seats available for this flight is:%d",list[check\_pt].no\_seats);  printf("\nEnter the number of seats to be reserved:");  scanf("%d",&seats);  list[check\_pt].no\_seats-=seats;  printf("\n\*\*\*\*\*\*\*\*Flight from %s to %s with %d seats is booked\*\*\*\*\*\*\*\*",list[check\_pt].start\_pt,list[check\_pt].des\_pt,seats);  }  else  {  printf("\nNo seats available.... Please try again later.");  }  printf("\nDo you want to make a another reservation press yes or no:");  scanf("%s",last\_check);  if(strcmp(last\_check,"yes")==0)  Customer\_input(list,num\_flight);  else  printf("\nTravel safe and hope that you have a great journey");  }  else  printf("No Flights found.");    }  void main()  {  int num\_flight;  printf("-------------WELCOME TO PANIC AIRLINES------------- ");  printf("Enter the number of flights to be recorded:");  scanf("%d",&num\_flight);  flight\_details(list,num\_flight);  Customer\_input(list,num\_flight);  } |
| **RESULT:** | |
| **INPUT:** |  |
| **OUTPUT:** |  |