|  |  |
| --- | --- |
| **Name** | Adwait S Purao |
| **UID no.** | 2021300101 |
| **Experiment No.** | 8 |

|  |  |
| --- | --- |
| **AIM:** | Apply the concepts of structures/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 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. |
| **ALGORITHM:** | 1. START  2. Define structure family with char array wife name, integer number of children and 2-D char array names as variables  3. Define union details with structure family and char array hobbies as variables  4. Define structure member with char array name, integer age, char array address characters active and married and union det as variables  5. Define void input function with member array c as variable  6. Loop from I = 0 to 1 Input all variables of members c[i] If c[i].married is equal to ‘Y’: Input all details of c[i].det.fam Else Input c[i].hobbies  7. Define function int main()  8. Declare variable c[i] of data type member  9. Call input(c)  10. Loop from I = 0 to 1 if c[i].married is equal to Y and c[i].active is equal to Y print c[i].name and c[i].det.fam.wife\_name  11. STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include<stdio.h>  typedef struct family  {  char name\_wife[20];  int numb\_child;  char name\_child[10][20];  }family;  typedef union details  {  family fam;  char hobby[30];  }details;  typedef struct member  {  char name[20];  int age;  char addr[100];  char active,married;  details det;  }member;  void input(member c[2])  {  for(int i=0;i<2;i++)  {  printf("\n\nEnter the name of the member: ");  scanf(" %s",c[i].name);  printf("Enter the age of the person: ");  scanf("%d",&c[i].age);  printf("Enter the address of the person: ");  scanf(" %[^\n]",c[i].addr);  printf("Enter active status (Y/N): ");  scanf(" %c",&c[i].active);  printf("Enter Marital Status (Y/N): ");  scanf(" %c",&c[i].married);  if(c[i].married=='Y')  {  printf("Enter name of the wife: ");  scanf(" %s",c[i].det.fam.name\_wife);  printf("Enter the number of children: ");  scanf("%d",&c[i].det.fam.numb\_child);  for(int i=0;i<c[i].det.fam.numb\_child;i++)  {  printf("Enter name of the children: ");  scanf(" %s",c[i].det.fam.name\_child[i]);  }  }  else  {  printf("Enter the hobby of the member: ");  scanf(" %[^\n]",c[i].det.hobby);  }  }  }  int main()  {  member c[2];  input(c);  printf("The list of married active players and their wives is as follows:\n");  printf("Name\tWife");  for(int i=0;i<2;i++)  {  if(c[i].active=='Y' && c[i].married=='Y')  printf("\n%s\t%s\n",c[i].name,c[i].det.fam.name\_wife);  }  return 0;  } |
| **RESULT:** | |
| **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). |
| **ALGORITHM:** | 1. START  2. Define structure airline\_t with char array src, dest, integers start, arrive, seats and counts as variables  3. Define void function reset with airline\_t variable c[]  4. Loop from I = 0 to 4 c[i].count is equal to 0  5. Define void function input with airline\_t variable c[]  6. Loop from I = 0 to 4 Input all details of c[i]  7. Define int main()  8. Initialize airline\_t variable c[5]  9. Call function input(c)  10. Do a. Input source and destination b. Flag = 0, D = 1 c. Loop from I = 0 to 4 If strcmp(c[i].src and source\_) is equal to 0 and if strcmp(c[i].dest and dest) I. print c[i].start, c[i].arrive and c[i].seats II. c[i].count = d III. d++ IV. temp++ d.If temp is equal to 0 print sorry no flights available else I. input choice and number of seat II. Loop from 0 to 4 if n is equal to c[i].count if c[i].seats – seat >=0 c[i].seats -= seat print Booked and remaining seats else print Seats not available e. Input flag f. call reset(c) while flag is equal to 0  11. Return 0  12. STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include<stdio.h>  #include<string.h>  typedef struct air  {  char src[4];  char des[4];  int start;  int arrival;  int seats;  int count;  }airline\_t;  void reset(airline\_t c[5])  {  for(int i=0;i<5;i++)  c[i].count=0;  }  void input(airline\_t c[5])  {  for(int i=0;i<5;i++)  {  printf("\nEnter the source: ");  scanf("%s",c[i].src);  printf("Enter the destination: ");  scanf("%s",c[i].des);  printf("Enter the starting time: ");  scanf("%d",&c[i].start);  printf("Enter the arriving time: ");  scanf("%d",&c[i].arrival);  printf("Enter the number of seats: ");  scanf("%d",&c[i].seats);  c[i].count=0;  }  }  int main()  {  int d=1,temp=0,n,flag=0,seat;  airline\_t c[5];  input(c);  char source[4], dest[4];  do  {  printf("\nEnter your source: ");  scanf("%s",source);  printf("Enter your destination: ");  scanf("%s",dest);  printf("\nStart\tEnd\tSeats\n");  for(int i=0;i<5;i++)  {  if(strcmp(c[i].src,source)==0 && strcmp(c[i].des,dest)==0 &&  c[i].seats>0)  {  printf("%d\t%d\t%d\n",c[i].start,c[i].arrival,c[i].seats);  c[i].count=d;  d++;  temp++;  }  }  if(temp==0)  {  printf("\nSorry we dont have any flights available");  }  else  {  printf("Enter the number of flight you want to take: ");  scanf("%d",&n);  printf("Enter the number of seats you want to book: ");  scanf("%d",&seat);  for(int i=0;i<5;i++)  {  if(n==c[i].count)  {  if(c[i].seats - seat >= 0)  {  c[i].seats -= seat;  printf("Your flight has been booked for %s to %s",c[i].src,c[i].des);  printf("\nTimings are follows: %d to %d",c[i].start,c[i].arrival);  printf("\nNo of seats remaining: %d",c[i].seats);  }  else  {  printf("The flight does not have %d seats available",seat);  printf("\nPlease select a different flight.");  }  }  }  }  printf("Enter 0 to continue booking or any other number to exit");  scanf("%d",&flag);  reset(c);  } while (flag==0);  return 0;  } |
| **RESULT:** | |
| **Program 3** | |
| **PROBLEM STATEMENT:** | A record in an organization’s payroll consists of one line for each employee consisting of: NAME (20 characters), GENDER (1 character M or F), SALARY (integer), DATE OF BIRTH (3 integers YEAR MONTH DAY). Write a program which will input 10 such records. Your program must then take in 5 amendments in the record set which will be in the same form as the record structure itself. The amendments can contain new employees to be added (name different from existing ones), employees left (salary given as 0) and update of salary(more or less). Your program must then incorporate these amendments and also remove those employees who have reached retirement age(Age 60). |
| **ALGORITHM:** | 1. START  2. Define structure employee with char array name, char gender, integers salary, day, month, year, age as variables  3. Define main function  4. Define employee variable c[10]  5. Define char array name[5][20]  6. Loop from I = 0 to 4 Input details of c[i] except age c[i].age = 2022 – c[i].year  7. Loop from k = 0 to 4  A. input name[k]  B. Loop from j = 0 to 9  I. if strcmp(c[i].name and name[k]) is equal to 0  a. input choice  b. if choice is equal to 1 c[j].salary is equal to 0 else input c[j].salary c. flag = 1 d. break  II. If flag is equal to 0  input all details of c[i]  c[i].age = 2022 – c[i].year  8. Loop from k = 0 to I  If(c[k].age is less than 60 and c[k].salary is greater than 0)  print all details of c[k]  9. Return 0  10. STOP |
| **FLOWCHART:** |  |
| **PROGRAM:** | #include<stdio.h>  #include<string.h>  typedef struct employee  {  char name[20];  char gender;  int salary;  int day;  int month;  int year;  int age;  }employee;  int main()  {  employee c[10];  int i;  char name[5][20];  for(i=0;i<5;i++)  {  printf("Enter the name: ");  scanf(" %s",c[i].name);  printf("Enter the gender (M/F): ");  scanf(" %c",&c[i].gender);  printf("Enter the salary: ");  scanf("%d",&c[i].salary);  printf("Enter the Date of Birth (DD MM YYYY): ");  scanf("%d %d %d",&c[i].day,&c[i].month,&c[i].year);  c[i].age = 2022 - c[i].year;  printf("\n");  }  printf("Amendments: ");  for (int k=0;k<5;k++)  {  int flag=0;  printf("\nEnter the name: ");  scanf(" %s",name[k]);  for(int j=0;j<10;j++)  {  if(strcmp(c[j].name,name[k])==0)  {  int choice;  printf("Enter 1 if employee has left and 2 if employee if the  salary is to be modified: ");  scanf("%d",&choice);  if(choice == 1)  c[j].salary = 0;  else if(choice == 2)  {  printf("Enter the new salary: ");  scanf("%d",&c[j].salary);  }  flag = 1;  break;  }  }  if(flag == 0)  {  strcpy(c[i].name,name[k]);  printf("Enter the gender of the employee (M/F): ");  scanf(" %c",&c[i].gender);  printf("Enter the salary of the employee: ");  scanf("%d",&c[i].salary);  printf("Enter the date of birth (DD MM YYYY) : ");  scanf("%d %d %d",&c[i].day,&c[i].month,&c[i].year);  c[i].age = 2022 - c[i].year;  i++;  }  }  printf("Name\tGender\tSalary\tDate of Birth");  for(int k=0;k<i;k++)  {  if(c[k].age <60 && c[k].salary != 0)  printf("\n%s\t%c\t%d\t%d %d  %d",c[k].name,c[k].gender,c[k].salary,c[k].day,c[k].month,c[k].year);  }  return 0;  } |
| **RESULT:** | |
| **Program 4** | |
| **PROBLEM STATEMENT:** |  |
| **ALGORITHM:** |  |
| **FLOWCHART:** |  |
| **PROGRAM:** |  |
| **RESULT:** | |
| **Program 5** | |
| **PROBLEM STATEMENT:** |  |
| **ALGORITHM:** |  |
| **FLOWCHART:** |  |
| **PROGRAM:** |  |
| **RESULT:** | |
| **CONCLUSION:** | We came to know that how we can use structures to store items of different type.We also learnt how to access the elements of the structures using the dot operator.We also learnt that structure can be called within a structure. |