LAB TASKS

**TASK 01**

**Create a structure to store details about cars in a dealership, including make,**

**model, year, price, and mileage. Write a program that allows users to add new**

**cars, display a list of available cars, and search for cars by make or model.**

**SOURCE CODE:**

#include<stdio.h>

#include<string.h>

struct car{

char make[15];

char model[15];

int year;

double price;

float mileage;

}c1[100];

int add\_cars(struct car c1[],int count)

{

printf("Enter car name:");

scanf(" %s",c1[count].make);

printf("Enter car model:");

scanf(" %s",c1[count].model);

printf("Enter year:");

scanf(" %d",&c1[count].year);

printf("Enter price:");

scanf("%lf",&c1[count].price);

printf("Enter mileage:");

scanf("%f",&c1[count].mileage);

return count+1;

}

void display\_cars(struct car c1[],int count)

{

if(count==0)

{

printf("No cars available\n");

return;

}

int i;

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

{

printf("Details of car %d\n",i+1);

printf("NAME:%s\n",c1[i].make);

printf("MODEL:%s\n",c1[i].model);

printf("YEAR:%d\n",c1[i].year);

printf("PRICE:%.2lf\n",c1[i].price);

printf("MILEAGE:%.2f\n",c1[i].mileage);

}

}

void search\_cars(struct car c1[],int count)

{

if(count==0)

{

printf("No cars available\n");

return;

}

char search[50];

printf("Enter car make or model to search:");

scanf("%s",search);

int i,f=0;

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

{

if(strcmp(search,c1[i].make)==0 ||strcmp(search,c1[i].model)==0)

{

f=1;

printf("Details of car %d\n",i+1);

printf("NAME:%s\n",c1[i].make);

printf("MODEL:%s\n",c1[i].model);

printf("YEAR:%d\n",c1[i].year);

printf("PRICE:%d\n",c1[i].price);

printf("MILEAGE:%f\n",c1[i].mileage);

}

if(!f)

printf("No car found with this make or model");

}

}

int main()

{

int choice;

int count = 0;

while(choice!=4)

{

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

printf("2.Search Car\n");

printf("3.Display Cars\n");

printf("4.Exit\n");

printf("Enter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1: count= add\_cars(c1,count);

break;

case 2: search\_cars(c1,count);

break;

case 3: display\_cars(c1,count);

break;

case 4: printf("Program terminated");

break;

default:

printf("Invalid choice");

}

}}

A screenshot of a computer

Description automatically generated

**TASK 02**

**Create a C program that defines a constant for the conversion factor of meters to**

**kilometers. Use a static variable in a function to count how many times the**

**function is called.**

**SOURCE CODE:**

#include<stdio.h>

#define factor 1000

float convert\_length(float length)

{

static int count = 0;

count++;

printf("Function called %d times\n",count);

return length/factor;

}

int main()

{

int count = 0;

float length;

while(1)

{

printf("Enter length in meters:");

scanf("%f",&length);

if(length<0)

break;

printf("Length in km:%.2f\n",convert\_length(length));

}

}

A screenshot of a computer program

Description automatically generated

**TASK 03**

**Write a recursive function linear Search that takes an array, its size, the target**

**element to search for, and the current index. It checks if the target is at the**

**current index and continues searching in the subsequent indices until it either**

**finds the target or exhausts the array.**

#include<stdio.h>

int linear\_search(int arr[],int size,int target,int i)

{

if(i>=size)

return -1;

if(arr[i]==target)

return i;

return linear\_search(arr,size,target,i+1);

}

int main()

{

int arr[5];

int target;

printf("Enter elements of array\n");

for(int i=0;i<5;i++)

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

int size = 5;

printf("Enter element to search:");

scanf("%d",&target);

int a = linear\_search(arr,size,target,0);

if(a==-1)

{

printf("Element not found in array");

}

else

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

}

A screenshot of a computer

Description automatically generated

**TASK 04**

**Write a recursive function to find if a subset of an array’s elements can sum to a**

**given target. For example, for array [3, 34, 4, 12, 5, 2] and target 9, the function**

**should return true.**

**SOURCE CODE:**

#include<stdio.h>

int check\_sum(int arr[],int n,int target)

{

if(target==0)

return 1;

if(n==0)

return 0;

if(arr[n-1]>target)

return check\_sum(arr,n-1,target);

else

return check\_sum(arr,n-1,target) || check\_sum(arr,n-1,target - arr[n-1]);

}

int main()

{

int target;

int arr[6];

printf("Enter 6 elements of array:\n");

for(int i =0;i<6;i++)

{

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

}

printf("Enter target element:");

scanf("%d",&target);

int result = check\_sum(arr,6,target);

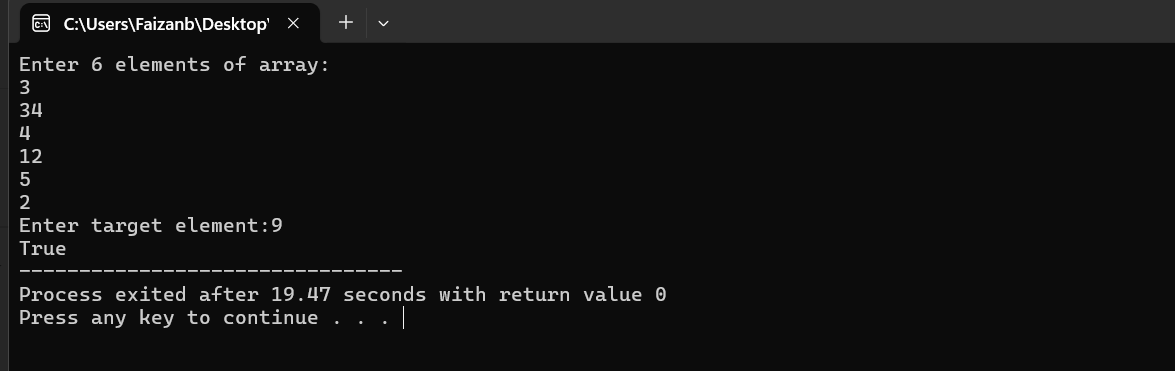
if(result == 1)

printf("True");

else

printf("False");

}



**TASK 05**

**Design a structure to store information about travel packages, including package**

**name, destination, duration, cost, and number of seats available. Write a program**

**that allows users to book a travel package and display available packages.**

**SOURCE CODE:**

#include<stdio.h>

#include<string.h>

struct travel\_packages{

int id;

char name[20];

char destination[20];

int duration;

int cost;

int seats;

};

void display\_packages(struct travel\_packages p1[],int count)

{

if(count == 0)

{

printf("No packages available to display\n");

return;

}

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

{

printf("Package %d details\n",i+1);

printf("Package id:%d\n",p1[i].id);

printf("Package name:%s\n",p1[i].name);

printf("Destination:%s\n",p1[i].destination);

printf("Duration:%d\n",p1[i].duration);

printf("Cost:%d\n",p1[i].cost);

printf("No of seats:%d\n",p1[i].seats);

}

}

void book\_packages(struct travel\_packages p1[],int count)

{

int package\_id;

int seats\_book;

printf("Enter package id to book:");

scanf("%d",&package\_id);

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

{

if(package\_id==p1[i].id)

{

printf("Enter no of seats to book:");

scanf("%d",&seats\_book);

if(seats\_book>0 && seats\_book<=p1[i].seats)

{

p1[i].seats-=seats\_book;

printf("%d seats booked successfully for package %d\n",seats\_book,package\_id);

}

else

{

printf("Invalid not of seats...");

}

return;

}

}

printf("Package not found....");

}

int main()

{

struct travel\_packages p1[3] = {

{1,"Mountain Escape", "Skardu", 6, 45000, 15},

{2,"Tropical Bliss", "Bali", 8, 200000, 8},

{3,"Desert Safari", "Dubai", 4, 120000, 12}

};

int choice;

while(choice!=3)

{

printf("1.Display package\n");

printf("2.Book a package\n");

printf("3.Exit\n");

printf("Enter your choice:");

scanf("%d",&choice);

switch(choice)

{

case 1 : display\_packages(p1,3);

break;

case 2 : book\_packages(p1,3);

break;

case 3: printf("Program terminated");

break;

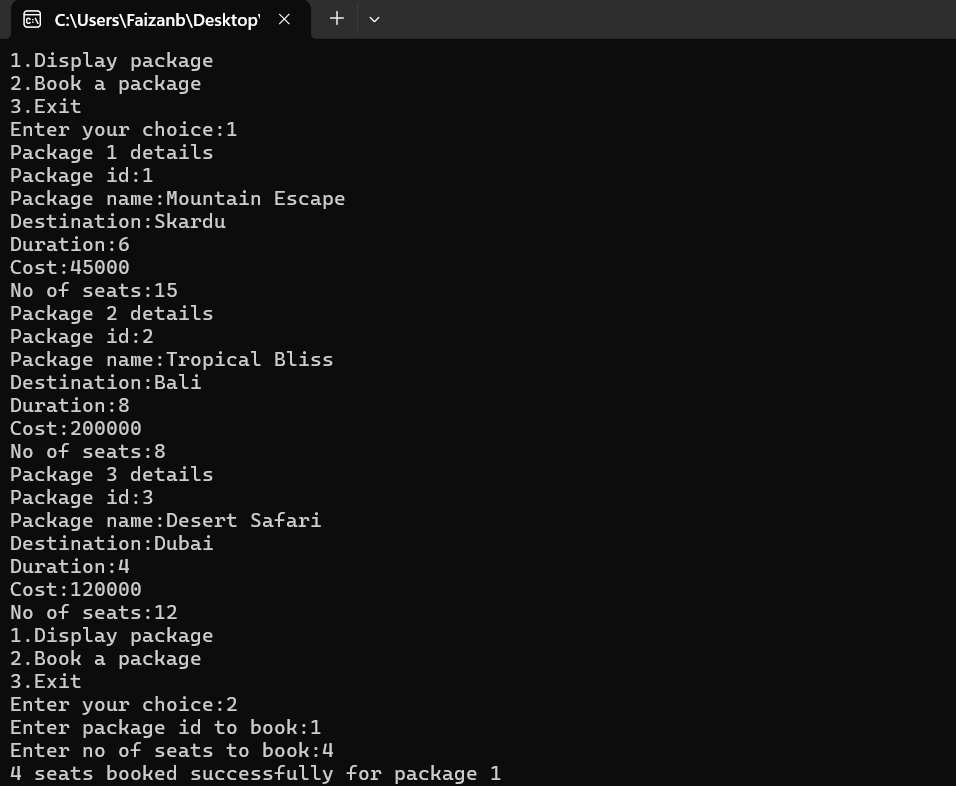
default:

printf("Invalid choice\n");

}

}

}



**TASK 06**

**Write a recursive function to count the number of vowels in a given string.**

**SOURCE CODE:**

#include<stdio.h>

#include<ctype.h>

int vowels\_count(char str[],int i)

{

if(str[i]=='\0')

return 0;

char ch = tolower(str[i]);

if(ch=='a' || ch == 'e' || ch=='i' || ch == 'o' || ch=='u')

{

return 1 + vowels\_count(str,i+1);

}

else

return vowels\_count(str,i+1);

}

int main()

{

char str[10];

printf("Enter a string:");

scanf("%[^\n]",str);

int count = vowels\_count(str,0);

printf("No of vowels:%d",count);

}

