**Data Structure**

**Lab-6**

**Submitted by: Submitted to:**

**Aakash Bhatt Pankaj Sir**

**500124633**

**(2023-2024)**

Q1.A record contains the name of a cricketer, his age, the number of test matches he has played, and the average runs he scored in each test match. Create an array of structures to hold records of 20 such cricketers and then write a program to read these records and arrange them in ascending order by runs, Use the qsort standard library function.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

*// Define a structure to represent cricketer data*

struct *cricketer*

{

    char name[50];

    int crick\_age;

    int match;

    float avg\_match;

};

*// Comparison function for qsort to compare cricketers based on average runs*

int compare(*const* void \*a, *const* void \*b)

{

*const* struct *cricketer* \*cricketerA = (*const* struct *cricketer* \*)a;

*const* struct *cricketer* \*cricketerB = (*const* struct *cricketer* \*)b;

    return (cricketerA->avg\_match > cricketerB->avg\_match) - (cricketerA->avg\_match < cricketerB->avg\_match);

}

int main()

{

    int i, n;

*// Prompt the user to enter the number of cricketers' data*

    printf("Enter the number of cricketers' data you want to insert: ");

    scanf("%d", &n);

*// Declare an array of structures to store cricketer data*

    struct *cricketer* obj1[20];

*// Input cricketer data from the user*

    for (i = 0; i < n; i++)

    {

        printf("Enter data of cricketer %d\n", i + 1);

        printf("Name: ");

        scanf("%s", obj1[i].name);

        printf("Age: ");

        scanf("%d", &obj1[i].crick\_age);

        printf("Matches: ");

        scanf("%d", &obj1[i].match);

        printf("Average runs: ");

        scanf("%f", &obj1[i].avg\_match);

    }

*// Sort the records using qsort and the compare function*

    qsort(obj1, n, sizeof(struct *cricketer*), compare);

*// Display the sorted records*

    printf("Sorted records:\n");

    for (i = 0; i < n; i++)

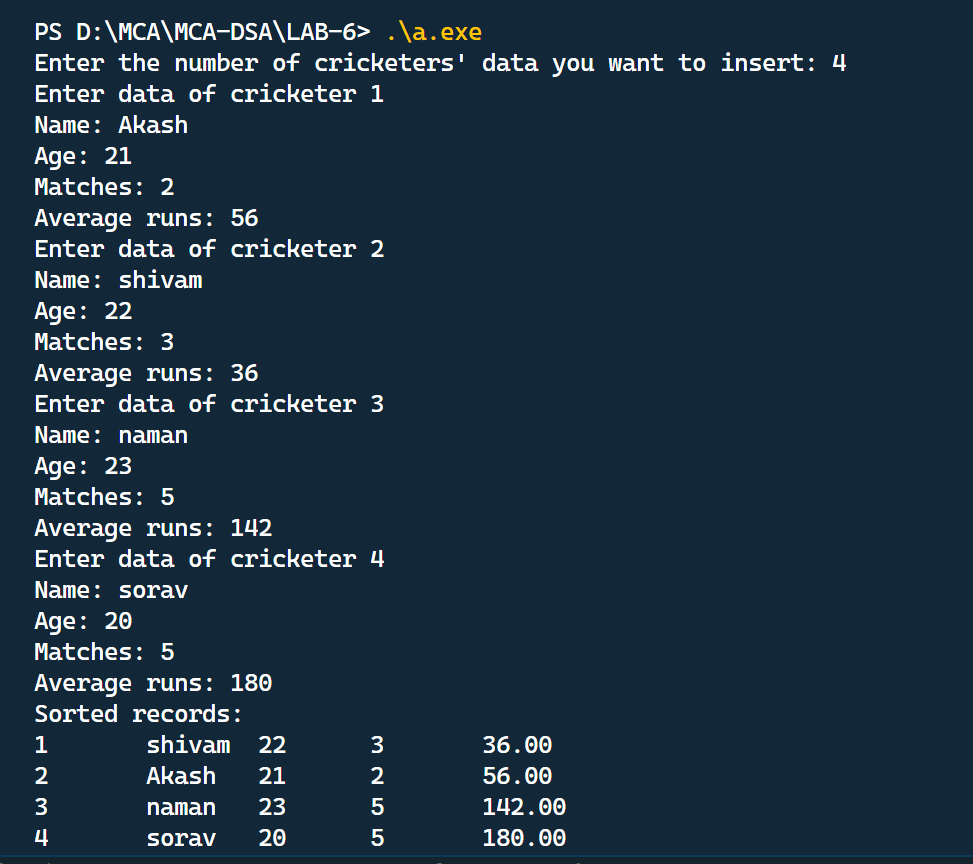
    {

        printf("%d\t%s\t%d\t%d\t%.2f\n", i + 1, obj1[i].name, obj1[i].crick\_age, obj1[i].match, obj1[i].avg\_match);

    }

    return 0;

}



2.Create a structure to specify data of customers in a bank. The data to be stored is Account number, Name, and Balance in the account. Assume a maximum of 200 customers in the bank.

#include <stdio.h>

#include <string.h>

*// Define the structure for customer data*

struct *Customer*

{

    int account\_number;

    char name[50];

    double balance;

};

int main()

{

*// Declare an array of Customer structures to store customer data*

    struct *Customer* customers[200];

    int num\_customers;

*// Prompt the user to enter the number of customers (up to 200)*

    printf("Enter the number of customers (up to 200): ");

    scanf("%d", &num\_customers);

*// Input customer data for each customer*

    for (int i = 0; i < num\_customers; i++)

    {

        printf("Customer #%d:\n", i + 1);

*// Input account number*

        printf("Account Number: ");

        scanf("%d", &customers[i].account\_number);

*// Input customer name (assuming single-word names)*

        printf("Name: ");

        scanf("%s", customers[i].name);

*// Input customer balance*

        printf("Balance: ");

        scanf("%lf", &customers[i].balance);

    }

*// Display customer data*

    printf("\nCustomer Data:\n");

    for (int i = 0; i < num\_customers; i++)

    {

        printf("Customer #%d:\n", i + 1);

        printf("Account Number: %d\n", customers[i].account\_number);

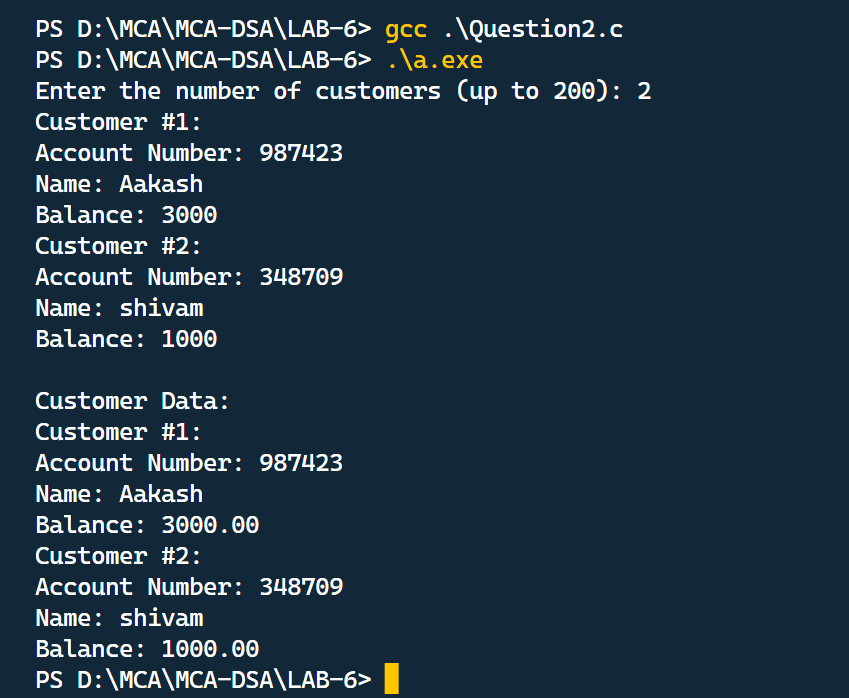
        printf("Name: %s\n", customers[i].name);

        printf("Balance: %.2lf\n", customers[i].balance);

    }

    return 0;

}



3.Write a function to print the account number and name of each customer with a balance below Rs 100.

#include <stdio.h>

#include <string.h>

*// Define the structure for customer data*

struct *Customer*

{

    int account\_number;

    char name[50];

    double balance;

};

*// Function to print customers with a balance below Rs 100*

void printCustomersBelow100(struct *Customer* customers*[]*, int num\_customers)

{

    printf("Customers with a balance below Rs 100:\n");

    for (int i = 0; i < num\_customers; i++)

    {

        if (customers[i].balance < 100.0)

        {

            printf("Account Number: %d\n", customers[i].account\_number);

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

            printf("Balance: %.2lf\n", customers[i].balance);

            printf("\n");

        }

    }

}

int main()

{

    struct *Customer* customers[200];

    int num\_customers;

*// Prompt the user to enter the number of customers*

    printf("Enter the number of customers: ");

    scanf("%d", &num\_customers);

*// Input customer data for each customer*

    for (int i = 0; i < num\_customers; i++)

    {

        printf("Customer #%d:\n", i + 1);

*// Input account number*

        printf("Account Number: ");

        scanf("%d", &customers[i].account\_number);

*// Input customer name (assuming single-word names)*

        printf("Name: ");

        scanf("%s", customers[i].name);

*// Input customer balance*

        printf("Balance: ");

        scanf("%lf", &customers[i].balance);

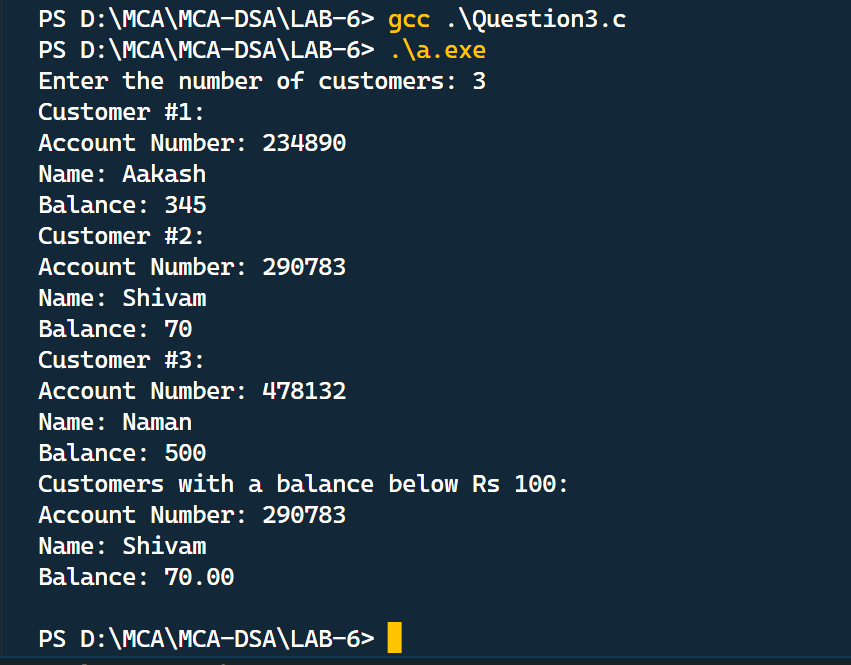
    }

*// Call the function to print customers with a balance below Rs 100*

    printCustomersBelow100(customers, num\_customers);

    return 0;

}



4.If a customer requests for withdrawal or deposit, the form contains the fields: Acct no, amount, code(1 for deposit and 0 for withdrawal) WAP to give a message " The balance is insufficient for the specified withdrawal", if on withdrawal the balance falls below Rs 100.

#include <stdio.h>

#include <string.h>

*// Define the structure for customer data*

struct *Customer*

{

    int account\_number;

    char name[50];

    double balance;

};

*// Function to perform a deposit or withdrawal transaction*

void performTransaction(struct *Customer* customers*[]*, int num\_customers, int acct\_no, double amount, int code)

{

    for (int i = 0; i < num\_customers; i++)

    {

        if (customers[i].account\_number == acct\_no)

        {

            if (code == 1)

            {

*// Deposit*

                customers[i].balance += amount;

                printf("Deposit of Rs %.2lf successful.\n", amount);

            }

            else if (code == 0)

            {

*// Withdrawal*

                if (customers[i].balance - amount < 100.0)

                {

                    printf("The balance is insufficient for the specified withdrawal.\n");

                }

                else

                {

                    customers[i].balance -= amount;

                    printf("Withdrawal of Rs %.2lf successful.\n", amount);

                }

            }

            else

            {

                printf("Invalid transaction code.\n");

            }

            return;

        }

    }

    printf("Account number %d not found.\n", acct\_no);

}

int main()

{

    struct *Customer* customers[200];

    int num\_customers;

*// Prompt the user to enter the number of customers*

    printf("Enter the number of customers: ");

    scanf("%d", &num\_customers);

*// Input customer data for each customer*

    for (int i = 0; i < num\_customers; i++)

    {

        printf("Customer #%d:\n", i + 1);

*// Input account number*

        printf("Account Number: ");

        scanf("%d", &customers[i].account\_number);

*// Input customer name (assuming single-word names)*

        printf("Name: ");

        scanf("%s", customers[i].name);

*// Input customer balance*

        printf("Balance: ");

        scanf("%lf", &customers[i].balance);

    }

    int acct\_no, code;

    double amount;

*// Prompt the user to enter transaction details*

    printf("Enter Account Number: ");

    scanf("%d", &acct\_no);

    printf("Enter Transaction Code (1 for Deposit, 0 for Withdrawal): ");

    scanf("%d", &code);

    printf("Enter Amount: ");

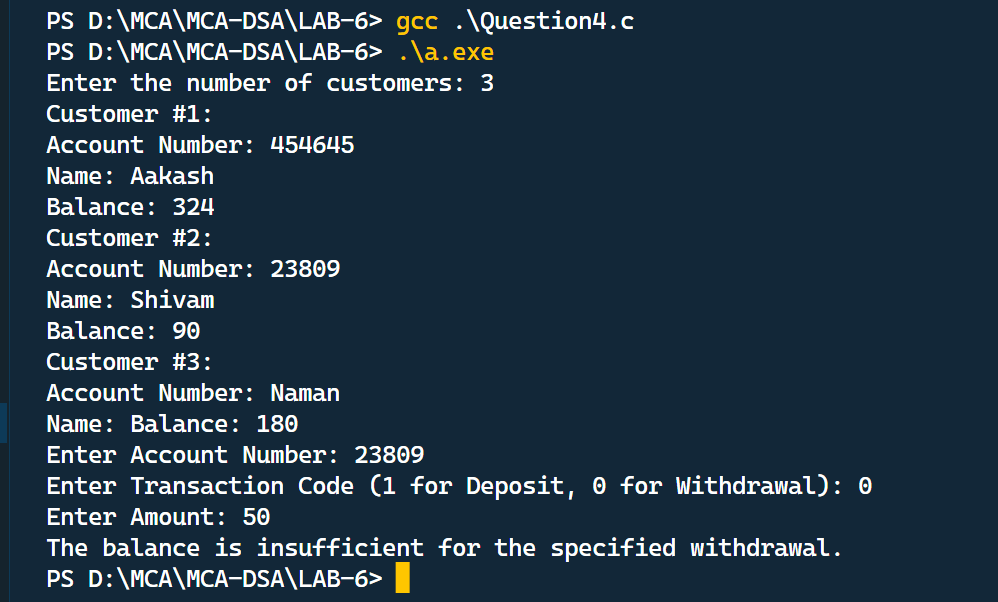
    scanf("%.2lf", &amount);

*// Call the function to perform the transaction*

    performTransaction(customers, num\_customers, acct\_no, amount, code);

    return 0;

}

****

5.WAP to count the number of occurrences of any two vowels in succession in a line of text. For example in the following sentence "Please read this application and give me gratuity". such occurrences, ea, ea and ui.

#include <stdio.h>

#include <string.h>

#include <stdbool.h>

*// Function to check if a character is a vowel*

bool isVowel(char c)

{

    switch (c)

    {

    case 'a':

    case 'e':

    case 'i':

    case 'o':

    case 'u':

    case 'A':

    case 'E':

    case 'I':

    case 'O':

    case 'U':

        return true;

    default:

        return false;

    }

}

int main()

{

    char text[1000];

*// Prompt the user to enter a line of text*

    printf("Enter a line of text: ");

    fgets(text, sizeof(text), stdin);

    int count = 0;

    int len = strlen(text);

*// Loop through the characters in the input text*

    for (int i = 0; i < len - 1; i++)

    {

*// Check if the current character and the next character are both vowels*

        if (isVowel(text[i]) && isVowel(text[i + 1]))

        {

*// If two vowels are found in succession, print them*

            printf("Found two vowels in succession: %c%c\n", text[i], text[i + 1]);

            count++;

        }

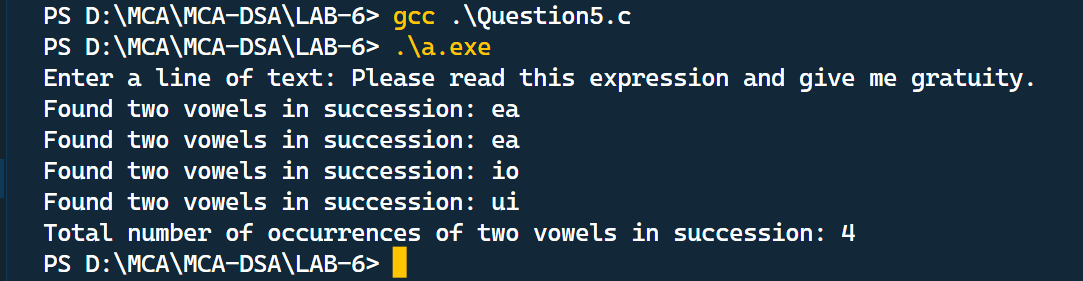
    }

*// Print the total number of occurrences of two vowels in succession*

    printf("Total number of occurrences of two vowels in succession: %d\n", count);

    return 0;

}



6.WAP to receive an integer and printout the number in words. For example, if the number is 5678, it should print Five thousand six hundred and seventy eight.

#include <stdio.h>

*// Function to print the word representation of a number (0-9)*

void printDigit(int digit)

{

*const* char \*words*[]* = {"Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine"};

    printf("%s ", words[digit]);

}

*// Function to convert a two-digit number into words*

void convertTwoDigits(int num)

{

    if (num < 10)

    {

        printDigit(num);

    }

    else if (num >= 10 && num <= 19)

    {

*const* char \*teens*[]* = {"Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen"};

        printf("%s ", teens[num - 10]);

    }

    else

    {

*const* char \*tens*[]* = {"", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety"};

        printf("%s ", tens[num / 10]);

        if (num % 10 > 0)

        {

            printDigit(num % 10);

        }

    }

}

*// Function to convert a three-digit number into words*

void convertThreeDigits(int num)

{

    if (num >= 100)

    {

        printDigit(num / 100);

        printf("Hundred ");

        num %= 100;

        if (num > 0)

        {

            printf("and ");

        }

    }

    convertTwoDigits(num);

}

int main()

{

    int number;

    printf("Enter an integer: ");

    scanf("%d", &number);

    if (number < 0)

    {

        printf("Negative ");

        number = -number;

    }

    if (number == 0)

    {

        printf("Zero\n");

    }

    else

    {

        if (number >= 1000)

        {

            convertThreeDigits(number / 1000);

            printf("Thousand ");

            number %= 1000;

        }

        convertThreeDigits(number);

        printf("\n");

    }

    return 0;

}

