Logo, company name

Description automatically generated

**İLERİ PROGRAMLAMA TEKNİKLERİ DERSİ FİNAL PROJESİ**

**Yusuf DEDE | Y210234103**

**SORU 1:** Aşağıdaki listenin gösterildiği bir program yazınız. Kullanıcıdan aşağıdaki listeden hangi işlemi yapmak istediğini sorunuz. Seçilen her bir işlemi ayrı bir fonksiyon içerisinde gerçekleştiriniz.

1. Fibonacci dizimi hesaplama
2. n!(n-1)!/(n+1)! hesaplama
3. Formül hesaplama

**Cevap;**

//  
// Created by Yusuf Dede on 29.05.2022.  
//  
#include <stdio.h>  
#include <stdlib.h>  
#include <stdbool.h>  
#include <ctype.h>  
#include <string.h>  
#include <math.h>  
  
const char menuDescription[] = "=============================================================\n"  
 "\t\t\t\t\t\t\t MENU\n"  
 "=============================================================\n\n"  
 "[ 1 ] -> Fibonacci dizimi hesaplama\n"  
 "[ 2 ] -> n!(n-1)!/(n+1)! hesaplama \n"  
 "[ 3 ] -> Formül hesaplama\n"  
 "[ -1 ] -> cikis..\n\n"  
 "==============================================================\n"  
 "Seciminizi giriniz: ";  
  
bool digitCheck(char key[]) {  
 for (int i = 0; i < strlen(key); i++) {  
 if (isdigit(key[i]) == 0) { // isdigit check int or char value  
 return false;  
 }  
 }  
 return true;  
}  
  
int faktoriyel(int sayi) {  
 if (sayi <= 1) return 1;  
 return sayi \* faktoriyel(sayi - 1);  
}  
  
void fibonacciSequenceCalculate() {  
 char oddOrEvenNumberinput[255];  
 int numberValue = 0;  
 int n1, n2;  
 n1 = 0;  
 n2 = 1;  
 int count = 0;  
  
 printf("Lütfen bir adet tam sayi giriniz: ");  
 scanf("%s", &oddOrEvenNumberinput);  
  
 while (!digitCheck(oddOrEvenNumberinput)) {  
 printf("Lutfen tam sayi olarak giriş yapiniz!! : ");  
 scanf("%s", &oddOrEvenNumberinput);  
 }  
 numberValue = atoi(oddOrEvenNumberinput); // char value covert to int  
  
 while (numberValue <= 0) {  
 printf("Lutfen POZITIF tam sayi olarak giriş yapiniz!! : ");  
 scanf("%s", &oddOrEvenNumberinput);  
 numberValue = atoi(oddOrEvenNumberinput); // char value covert to int  
 }  
  
 if (numberValue == 1) {  
 printf("\n\nFibonacci dizisi kadar %d : %d\n\n", numberValue, n1);  
 } else {  
 while (count < numberValue) {  
 printf("%d\n", n1);  
 int nth = n1 + n2;  
 n1 = n2;  
 n2 = nth;  
 count += 1;  
 }  
 }  
}  
  
void factorialCalculate() {  
 char oddOrEvenNumberinput[255];  
 int numberValue;  
  
 printf("Lütfen bir adet tam sayi giriniz: ");  
 scanf("%s", &oddOrEvenNumberinput);  
  
 while (!digitCheck(oddOrEvenNumberinput)) {  
 printf("Lutfen tam sayi olarak giriş yapiniz!! : ");  
 scanf("%s", &oddOrEvenNumberinput);  
 }  
 numberValue = atoi(oddOrEvenNumberinput); // char value covert to int  
  
 int temp = 1;  
 for (int i = 1; i < numberValue; i++) {  
 temp = i \* temp;  
 }  
 printf("Sonuc: %d\n", temp / numberValue + 1);  
}  
  
void formulaCalculate() {  
 char oddOrEvenNumberinput[255];  
 int numberValue = 0;  
 int n[3];  
  
  
 for (int i = 0; i < 3; ++i) {  
 if (i == 0) { printf("a: Lütfen bir adet tam sayi giriniz: "); }  
 else if (i == 1) { printf("b: Lütfen bir adet tam sayi giriniz: "); }  
 else if (i == 2) { printf("c: Lütfen bir adet tam sayi giriniz: "); }  
 scanf("%s", &oddOrEvenNumberinput);  
 while (!digitCheck(oddOrEvenNumberinput)) {  
 printf("Lutfen tam sayi olarak giriş yapiniz!! : ");  
 scanf("%s", &oddOrEvenNumberinput);  
 }  
 numberValue = atoi(oddOrEvenNumberinput); // char value covert to int  
 n[i] = numberValue;  
 }  
  
 int a=n[0], b=n[1], c=n[2];  
 int temp = b \* b - 4 \* a \* c;  
 int sqrt\_result = sqrt(temp);  
 temp = sqrt\_result - b;  
 printf("Formul Hesap Sonucu: %d\n", temp / 2 \* b);  
}  
  
void menu() {  
 int choice = 0;  
  
 while (choice != 4) {  
 fflush(stdin);  
  
 printf("%s", menuDescription);  
 scanf("%d", &choice);  
  
 switch (choice) {  
 case 1:  
 fibonacciSequenceCalculate();  
 choice = 0;  
 break;  
 case 2:  
 factorialCalculate();  
 choice = 0;  
 break;  
 case 3:  
 formulaCalculate();  
 choice = 0;  
 break;  
 case -1:  
 printf("Programdan cikiş yapildi!");  
 exit(1);  
 default:  
 printf("\nLuften seceneklerden birini seciniz !!!!!!\n\n");  
 choice = 0;  
 break;  
 }  
 }  
}

int main(int argc, char \*argv[]) {  
 menu();  
 return 0;  
}

**SORU 2:** Bir sınıfta 10 öğrenci bulunmaktadır ve öğrenciler matematik ve fizik derslerinden not almaktadırlar. Kullanıcıdan öğrencilerin matematik ve fizik notlarını alarak çok boyutlu bir dizide saklayınız. Hesaplamalar: **(35p)**

* Fizik ve matematik dersi not ortalaması hesaplayınız.
* Fizik dersi notlarını azalan sıra ile yazınız.
* Fizik dersinden en yüksek not alan öğrencinin matematik dersi notunu yazınız.
* Matematik not ortalamasına en yakın notu olan öğrencinin Matematik notunu yazınız.

**Cevap;**

//  
// Created by Yusuf Dede on 2.06.2022.  
//  
#include <stdio.h>  
#include <stdlib.h>  
  
/\* BASE DEFINITIONS - BEGIN \*/  
  
#define ROW\_COUNT 5 // student count  
#define COLUMN\_COUNT 2 // physics & math column  
#define PHYSICS 1  
#define MATH 2  
  
/\* BASE DEFINITIONS - END \*/  
  
  
int binarySearch(int nums[], int n, int target) {  
 int low = 0;  
 int high = n - 1;  
  
 while (low <= high) {  
 int mid = low + (high - low) / 2;  
  
 if (nums[mid] < target) {  
 low = mid + 1;  
 } else if (nums[mid] > target) {  
 high = mid - 1;  
 } else {  
 return mid;  
 }  
 }  
  
 return low;  
}  
  
void findKClosestElements(int nums[], int target, int n) {  
 int i = binarySearch(nums, n, target);  
 int k = 3;  
  
 int left = i - 1;  
 int right = i;  
  
 while (k-- > 0) {  
 if (left < 0 || (right < n && abs(nums[left] - target) > abs(nums[right] - target))) {  
 right++;  
 } else {  
 left--;  
 }  
 }  
  
 int highCloseScore = 0;  
  
 left++;  
 while (left < right) {  
 highCloseScore = nums[left];  
 break;  
 left++;  
 }  
  
 printf("Ortalamaya en yakin ogrencinin matamatik notu: %d\n", highCloseScore);  
}  
  
void isPhysicsOrMathInputPrint(int index) {  
 if (index == PHYSICS) { printf("Ogrenci Fizik Notu : "); }  
 else if (index == MATH) { printf("Ogrenci Matematik Notu : "); }  
}  
  
void getInputPhysicsMath(int examScores[ROW\_COUNT][COLUMN\_COUNT]) {  
 int row, column, index;  
 for (row = 0; row < ROW\_COUNT; row++) {  
 index = 0;  
 printf("%d. Ogrenci\n\n", row + 1);  
 for (column = 0; column < COLUMN\_COUNT; column++) {  
 index += 1;  
 isPhysicsOrMathInputPrint(index);  
 scanf("%d", &examScores[row][column]);  
 }  
 printf("\n");  
 }  
}  
  
int printScores(int scores[ROW\_COUNT]) {  
 int averageResult = 0;  
 for (int i = 0; i < ROW\_COUNT; ++i) {  
 averageResult += scores[i] / ROW\_COUNT;  
 printf("%d", scores[i]);  
 if (i == ROW\_COUNT - 1) { printf("]"); }  
 else { printf(", "); }  
 }  
 return averageResult;  
}  
  
void printSortingScores(int scores[ROW\_COUNT]) {  
 int temp;  
 for (int i = 0; i < ROW\_COUNT; ++i) {  
 for (int j = i + 1; j < ROW\_COUNT; ++j) {  
 if (scores[i] < scores[j]) {  
 temp = scores[i];  
 scores[i] = scores[j];  
 scores[j] = temp;  
 }  
 }  
 }  
 printScores(scores);  
}  
  
int printFindHighPhysicsMathScore(int scores[ROW\_COUNT]) {  
 int \*tempScores;  
 tempScores = scores;  
  
 int index = ROW\_COUNT;  
 for (int i = 1; i < ROW\_COUNT; ++i) {  
  
 if (tempScores[0] < tempScores[i]) {  
 tempScores[0] = tempScores[i];  
 index -= 1;  
 }  
  
 }  
 printf("Fizik Dersinden En yuksek Not: %d", tempScores[0]);  
 return index;  
}  
  
  
void calculateOptionsExecute(int examScores[ROW\_COUNT][COLUMN\_COUNT]) {  
 int physicsScores[ROW\_COUNT];  
 int mathScores[ROW\_COUNT];  
 int originalMathScores[ROW\_COUNT];  
  
 int averagePhysicsResult, averageMathResult;  
  
 int row, column, index;  
  
 for (row = 0; row < ROW\_COUNT; row++) {  
 index = 0;  
 for (column = 0; column < ROW\_COUNT; column++) {  
 index += 1;  
 if (index == PHYSICS) { physicsScores[row] = examScores[row][column]; }  
 else if (index == MATH) {  
 mathScores[row] = examScores[row][column];  
 originalMathScores[row] = examScores[row][column];  
 }  
 }  
 }  
  
 printf("========================================================================\n"  
 "Fizik & Matematik Dersi Not Ortalamasi Sonuclari\n"  
 "========================================================================\n\n");  
 printf("Fizik ders notları: [");  
 averagePhysicsResult = printScores(physicsScores);  
 printf(" - Fizik Ortalama: %d\n", averagePhysicsResult);  
  
 printf("Matematik ders notları: [");  
 averageMathResult = printScores(mathScores);  
 printf(" - Matematik Ortalama: %d\n\n", averageMathResult);  
  
  
 printf("========================================================================\n"  
 "Fizik & Matematik Dersi Notlarinin Azalan Sira Sonuclari\n"  
 "========================================================================\n\n");  
 printf("Fizik ders notları azalan sira: [");  
 printSortingScores(physicsScores);  
 printf("\n");  
  
 printf("Matematik ders notlari azalan sira: [");  
 printSortingScores(mathScores);  
 printf("\n");  
  
 printf("\n========================================================================\n"  
 "Fizik dersinden en yuksek not alan ogrencinin matematik dersi notu\n"  
 "========================================================================\n\n");  
  
 int highPhysicsScoreStudentIndex = printFindHighPhysicsMathScore(physicsScores);  
 printf("\nOgrencinin Matematik Notu: %d\n", originalMathScores[highPhysicsScoreStudentIndex]);  
  
 printf("\n========================================================================\n"  
 "Matematik not ortalamasina en yakın notu olan ogrencinin Matematik notu\n"  
 "========================================================================\n\n");  
  
 printf("Matamatik not otalamasi: %d\n", averageMathResult);  
 findKClosestElements(mathScores, averageMathResult, ROW\_COUNT);  
}  
  
  
int main(int argc, char \*argv[]) {  
 int examScores[ROW\_COUNT][COLUMN\_COUNT] ;  
 getInputPhysicsMath(examScores);  
 calculateOptionsExecute(examScores);  
}

**SORU 3:** Kendi belirlemiş olduğunuz bir senaryo üzerinden Dosya açma, Dosyaya veri ekleme, Dosyadan veri çekip ekranda gösterme uygulamaları içeren bir program yazınız. **(30p)**

* **Telefon Rehberi**

**Cevap;**

//  
// Created by Yusuf Dede on 3.06.2022.  
//  
#include <stdio.h>  
#include <stdlib.h>  
#include <locale.h>  
#include <string.h>  
#include "conio.h"  
  
typedef struct {  
 char name[100];  
 long int code;  
 long int number;  
} telephone;  
  
int search, status = 1, i, c;  
char search1[50];  
  
  
FILE \*kp;  
telephone record;  
  
void delete(void);  
void edit(void);  
  
void delete(void) {  
 int silno;  
 char siltekrar, ch;  
  
 printf("\nSilinecek Musteri Kodu : ");  
 scanf("%d", &silno);  
  
 fseek(kp, silno \* sizeof(telephone), 0);  
 fread(&record, sizeof(telephone), 1, kp);  
 if (record.code == silno) {  
 printf("\nMusteri Kodu : %d", record.code);  
 printf("\nMusteri Adi : %s", record.name);  
 printf("\nTelefon : ");  
 printf("%d", record.number);  
 printf("\nKAYIT SILINICEK!...<E/H>");  
 ch = c\_getch();  
 }  
  
 if (ch == 'E' || ch == 'e') {  
 strcpy(record.name, "");  
 record.number = 0.0;  
 fseek(kp, silno \* sizeof(record), 0);  
 fwrite(&record, sizeof(record), 1, kp);  
 printf("\nKAYIT BASARIYLA SILINDI\n\n");  
 } else {  
 if (record.code != silno) {  
 printf("KAYIT BULANAMADI!...");  
 }  
 printf("\n\nTekrar Arama Yapmak istermisiniz <E/H> ");  
 siltekrar = c\_getch();  
 if (siltekrar == 'E' || siltekrar == 'e') {  
 printf("\nSilinecek Musteri Kodu : ");  
 scanf("%d", &silno);  
  
 fseek(kp, silno \* sizeof(telephone), 0);  
 fread(&record, sizeof(telephone), 1, kp);  
 if (record.code == silno) {  
 printf("\nMusteri Kodu : %d", record.code);  
 printf("\nMusteri Adi : %s", record.name);  
 printf("\nTelefon : ");  
 printf("%d", record.number);  
 printf("\nKAYIT SILINICEK!...<E/H>");  
 ch = c\_getch();  
 }  
 if (ch == 'E' || ch == 'e') {  
 strcpy(record.name, "");  
 record.number = 0.0;  
 fseek(kp, silno \* sizeof(record), 0);  
 fwrite(&record, sizeof(record), 1, kp);  
 printf("\nKAYIT BASARIYLA SILINDI\n\n");  
 }  
 if (record.code != silno) {  
 printf("KAYIT BULANAMADI!...");  
 }  
 }  
  
 }  
 printf("\n\n Devam icin herhangi bir tusa basin");  
 c\_getch();  
}  
  
void recordAdd() {  
 printf("\nMusteri Kodu :");  
 scanf("%d", &record.code);  
 printf("Musteri Adi :");  
 scanf("%s", &record.name);  
 printf("Telefon :");  
 scanf("%d", &record.number);  
  
 fseek(kp, record.code \* sizeof(telephone), 0); // Isaretciyi en basa cek SEEK\_SET  
 fwrite(&record, sizeof(telephone), 1, kp);  
 fflush(kp);  
 //fflush(FILE \*) tampon bellektekiler diske yazilacaktir....  
}  
  
void list() {  
 system("CLS");//clrscr();  
  
 fseek(kp, 0, 0);  
 fread(&record, sizeof(telephone), 1, kp);  
 printf("\nM.Kodu Adi Telefon\n==========================\n");  
 while (!feof(kp)) {  
 if (strcmp(record.name, "")) {  
 printf("\n%ld \t %s \t ", record.code, record.name);  
 printf("%ld", record.number);  
 }  
 fread(&record, sizeof(telephone), 1, kp);  
 }  
 printf("\n\n Devam icin herhangi bir tusa basin");  
 c\_getch();  
}  
  
void nameSearch() {  
 printf("\nAranacak Isim : ");  
 scanf("%s", &search1);  
  
 fseek(kp, 0, 0);  
 fread(&record, sizeof(record), 1, kp);  
 while (!feof(kp)) {  
 if (!strncmp(record.name, search1, strlen(search1))) {  
 printf("\nMusteri Kodu : %d", record.code);  
 printf("\nMusteri Adi : %s", record.name);  
 printf("\nTelefon : %d", record.number);  
 status = 0;  
 }  
 fread(&record, sizeof(record), 1, kp);  
 }  
 if (status) {  
 printf("\nKAYIT BULUNAMADI");  
 }  
 printf("\n\n Devam icin herhangi bir tusa basin");  
 c\_getch();  
}  
  
void numberSearch() {  
 printf("\nAranacak Musteri Numarasi : ");  
 scanf("%d", &search);  
  
 fseek(kp, 0, 0);  
 fread(&record, sizeof(telephone), 1, kp);  
  
 while (!feof(kp)) {  
 if (record.number == search) {  
 printf("\nMusteri Kodu: %d", record.code);  
 printf("\nMusteri Adi : %s", record.name);  
 printf("\nTelefon : %d", record.number);  
 break;  
  
 }  
 fread(&record, sizeof(telephone), 1, kp);  
 }  
 if (record.number != search) {  
 printf("\nKAYIT BULUNAMADI");  
 }  
  
 printf("\n\n Devam icin herhangi bir tusa basin");  
 c\_getch();  
  
}  
  
void edit(void) {  
 int dno;  
 char tekrar, ch;  
 printf("\nDuzeltilecek Musteri Kodu : ");  
 scanf("%d", &dno);  
 fseek(kp, dno \* sizeof(record), 0);  
 fread(&record, sizeof(record), 1, kp);  
 if (record.code == dno) {  
 printf("\nMusteri Kodu: %d", record.code);  
 printf("\nMusteri Adi : %s", record.name);  
 printf("\nTelefon : %d", record.number);  
 printf("\nDUZELTILECEK KAYIT BU MU? <E/H>");  
 ch = c\_getch();  
 }  
  
 if (ch == 'E' || ch == 'e') {  
 printf("\nMusteri Adi: ");  
 scanf("%s", record.name);  
 printf("\nTelefon : ");  
 scanf("%d", &record.number);  
 fseek(kp, dno \* sizeof(record), 0);  
 fwrite(&record, sizeof(record), 1, kp);  
 printf("\nKAYIT BASARIYLA DUZELTILDI\n\n");  
 } else {  
 if (record.code != dno) {  
 printf("KAYIT BULANAMADI!...");  
 }  
 printf("\n\nTekrar Arama Yapmak isterminiz <E/H> ");  
 tekrar = c\_getch();  
 if (tekrar == 'E' || tekrar == 'e') {  
 printf("\nDuzeltilecek Musteri Kodu : ");  
 scanf("%d", &dno);  
  
 fseek(kp, dno \* sizeof(record), 0);  
 fread(&record, sizeof(record), 1, kp);  
 if (record.code == dno) {  
 printf("\nMusteri Kodu: %d", record.code);  
 printf("\nMusteri Adi : %s", record.name);  
 printf("\nTelefon : %d", record.number);  
 printf("\nDUZELTILECEK KAYIT BU MU? <E/H>");  
 ch = c\_getch();  
 }  
 if (ch == 'E' || ch == 'e') {  
 printf("\nMusteri Adi : ");  
 scanf("%s", record.name);  
 printf("\nTelefon : ");  
 scanf("%d", &record.number);  
 fseek(kp, dno \* sizeof(record), 0);  
 fwrite(&record, sizeof(record), 1, kp);  
 printf("\nKAYIT BASARIYLA DUZELTILDI\n\n");  
 }  
 if (record.code != dno) {  
 printf("KAYIT BULANAMADI!...");  
 }  
 } else {  
 printf("KAYIT DUZELTILMEDI!... ");  
 }  
 }  
 printf("\n\n Devam icin herhangi bir tusa basin");  
 c\_getch();  
}  
  
int main() {  
  
 setlocale(LC\_ALL, "turkish");  
  
 kp = fopen("TelefonRehber.txt", "r+");  
  
 if (kp == NULL)  
 kp = fopen("TelefonRehber.txt", "w+");  
 do {  
 system("CLS"); // Ekranı temizler...  
 printf("\n\t\t TELEFON REHBERI v1.0");  
 printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");  
 printf("ISLEMLER\n\n");  
 printf("1 - Kayit Ekle\n");  
 printf("2 - Telefonlari list\n");  
 printf("3 - Kaydi Duzenle\n");  
 printf("4 - Numara ile Kayit bul\n");  
 printf("5 - Isim ile Kayit Bul\n");  
 printf("6 - Kayit Sil\n");  
 printf("0 - Cikis\n");  
  
 printf("\n\n\n Secim... ");  
 scanf("%d", &c);  
  
  
 if (c == 1) {  
 recordAdd();  
 } else if (c == 2) {  
 list();  
 } else if (c == 3) {  
 edit();  
 } else if (c == 4) {  
 numberSearch();  
 } else if (c == 5) {  
 nameSearch();  
 } else if (c == 6) {  
 delete();  
 }  
 } while (c != 0);  
 fclose(kp);  
}

TEŞEKKÜRLER..

Kaynak kod ektedir..