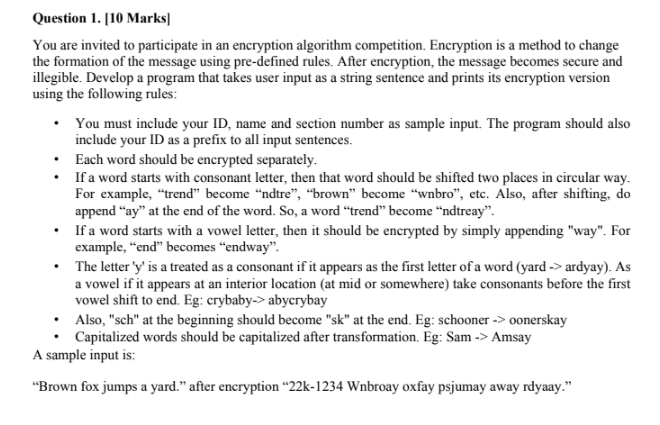
Assignment 3

Laiba Fatima 22k-5195



#include<stdio.h>

#include<string.h>

#include<ctype.h>

void Encrypt(){

char Input[100];

char Code[100][100];

char TempWord[16];

char Temp;

int i,j=0,k,l,m,o,len=0,len2=0,index=0,F=0,A,W=0;

char StudentID[20];

char Name[40];

char Section[10];

printf("\n\nEnter your Student ID: ");

gets(StudentID);

printf("Enter your Name: ");

gets(Name);

printf("Enter section number: ");

scanf("%s",Section);

printf("Enter the string to be encrypted: ");

getchar();

gets(Input);

len= strlen(Input);

Input[len]='\0';

for(i=0;i<len+1;i++){

if (Input[i]==' '|| Input[i]=='\0'){

A=0;

for(;j<i;j++){

TempWord[A]=Input[j];

A++;

}

W++;

j=i+1;

len2=strlen(TempWord);

//vowel condition

if(TempWord[0]=='A'||TempWord[0]=='a' || TempWord[0]=='E'||TempWord[0]=='e' || TempWord[0]=='I'||TempWord[0]=='i' ||

TempWord[0]=='O'||TempWord[0]=='o' || TempWord[0]=='U'||TempWord[0]=='u'){

TempWord[len2] ='w';

TempWord[len2+1] ='a';

TempWord[len2+2] ='y';

}

else{

//normal consonant

if(TempWord[0]!='y'||TempWord[0]!='Y'){

for(k=1;k<len2;k++){

//checking if word contains y

if(TempWord[k]=='y'||TempWord[k]=='Y'){

for(o=0;o<len2;o++){

//checking if there is a vowel after y

if((((TempWord[o]=='A'||TempWord[o]=='a') ||

(TempWord[o]=='E'||TempWord[o]=='e')) ||

((TempWord[o]=='I'||TempWord[o]=='i') ||

(TempWord[o]=='O'||TempWord[o]=='o')))||

(TempWord[o]=='U'||TempWord[o]=='u')){

index = len2-o;

for(l=0;l<index;l++){

Temp=TempWord[len2-1];

for(m=len2-1;m>0;m--){

TempWord[m]=TempWord[m-1];

}

TempWord[0]= Temp;

}

TempWord[len2] ='a';

TempWord[len2+1]='y';

F=1;

}

else{

F = 0;

continue;

}

break;

}

break;

}

else{

continue;

}

}

}

else{

F = 0;

}

if(F==0){ //applying the sch condition

if(TempWord[0]=='s'||TempWord[0]=='S'){

if(TempWord[1]=='c'){

if(TempWord[2]=='h'){

TempWord[1]='k';

for(k=3;k<len2;k++){

TempWord[k-1]=TempWord[k];

}

TempWord[len2-1]='\0';

for(l=0;l<len2-2-1;l++){

Temp=TempWord[len2-2];

for(m=len2-2;m>0;m--){

TempWord[m]=TempWord[m-1];

}

TempWord[0]= Temp;

}

TempWord[len2-1] ='a';

TempWord[len2]='y';

}

else{

for(l=0;l<2;l++){

Temp=TempWord[len2-1];

for(m=len2-1;m>0;m--){

TempWord[m]=TempWord[m-1];

}

TempWord[0]= Temp;

}

TempWord[len2] ='a';

TempWord[len2+1]='y';

}

}

else{

for(l=0;l<2;l++){

Temp=TempWord[len2-1];

for(m=len2-1;m>0;m--){

TempWord[m]=TempWord[m-1];

}

TempWord[0]= Temp;

}

TempWord[len2] ='a';

TempWord[len2+1]='y';

}

}

else{

for(l=0;l<2;l++){

Temp=TempWord[len2-1];

for(m=len2-1;m>0;m--){

TempWord[m]=TempWord[m-1];

}

TempWord[0]= Temp;

}

TempWord[len2] ='a';

TempWord[len2+1]='y';

}

}

}//checking upper and lower case

for(l=0;l<len2+4;l++){

TempWord[l]=tolower(TempWord[l]);

}

if(W==1){

TempWord[0]=toupper(TempWord[0]);

}

strcpy(Code[0],StudentID);

for(k=W;k<W+1;k++){

strcpy(Code[k],TempWord);

}

for(k=0;k<len2+4;k++){

TempWord[k]='\0';

}

}

else{

continue;

}

}

printf("\nYour encrypted code is: \n\t");

for(i=0;i<W+1;i++){

printf("%s ",Code[i]);

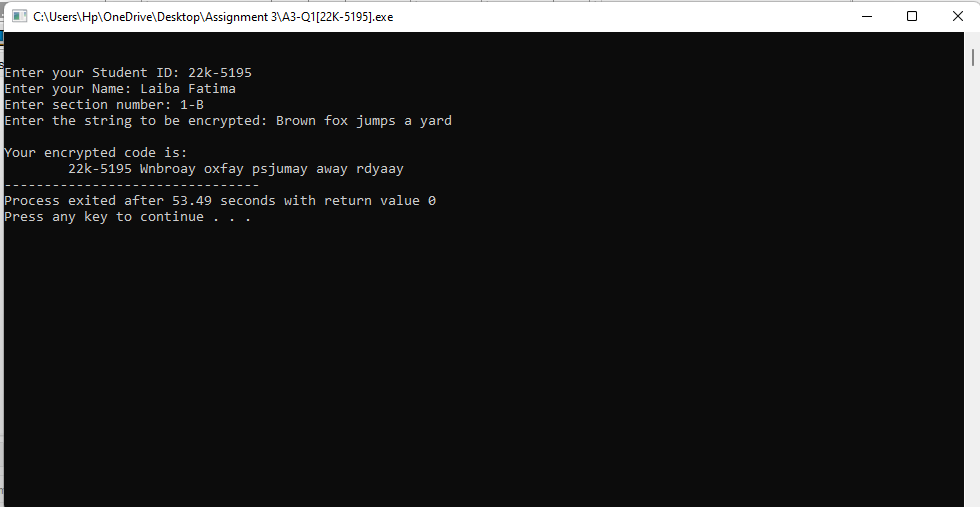
}

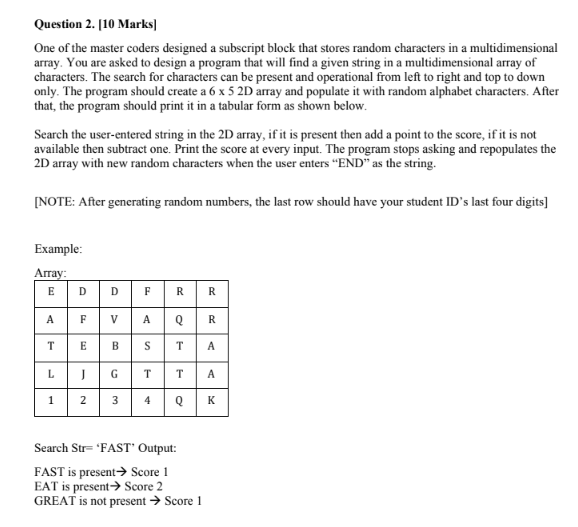
}

int main(){

Encrypt();

}





#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void find(char[5][6]);

main(){

char id[8],name[20];

char array[5][6];

int i,j,k=4;

printf ("Enter your name=");

gets(name);

printf ("Enter your id=");

gets(id);

printf ("Array:\n");

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

for(j=0;j<6;j++){

if(i==4 && j>0){

array[i][j]=id[k];

k++;

printf ("%c ",array[i][j]);

}

else{

array[i][j]=(rand()%26)+65;

printf ("%c ",(array[i][j]));

}

}

printf ("\n");

}

find(array);

}

void find(char arr[5][6]){

int r,i,count=0,j,k,score=0,s,n=0,v,b=0,d;

char str[30],end[3]={'E','N','D'};

a:

printf ("Enter string you want to search= ");

gets(str);

count=0;

b=0;

s=strlen(str);

r=strcmp(str,end);

if(r==0)

exit(0);

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

for(j=0;j<6;j++){

b=0;

if(arr[i][j]==str[b]){

b++;

count++;

if(arr[i][j+b]==str[b]){

b++;

count++;

for(d=b;str[d]!='\0';d++){

if(arr[i][j+d]==str[d]){

count++;

b++;

}

}

}

else if(arr[i+b][j]==str[b]){

b++;

count++;

for(d=b;str[d]!='\0';d++){

if(arr[i+d][j]==str[d]){

count++;

b++;

}

}

}

else{

b=0;

count=0;

}

}

if(count==s){

printf ("string is present\n");

score++;

printf ("your score is %d\n",score);

goto a;

}

}

}

if(count!=s){

printf ("string is not present\n");

score--;

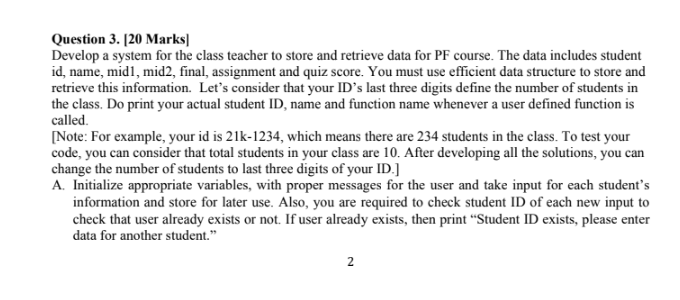
printf ("your score is %d\n",score);

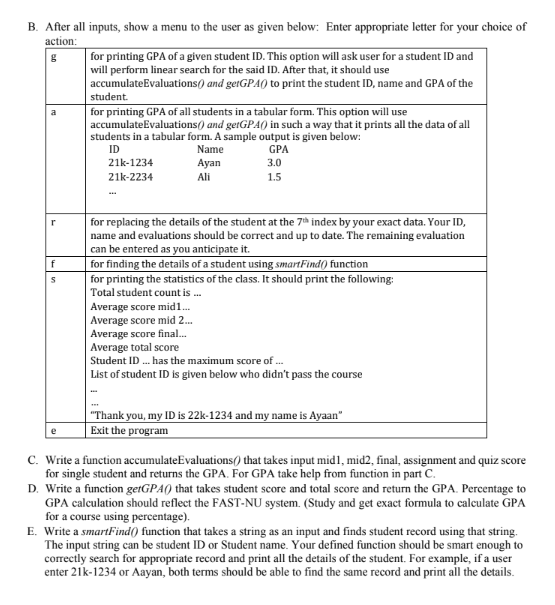
goto a;

}

}







#include<string.h>

#include<stdio.h>

struct details{

char StudentID[20];

char Name[50];

int Mid1;

int Mid2;

int Quiz;

int Assignment;

int Final;

int Total;

float GPA;

};

void smartFind(struct details Students){

printf("\n\nName: %s", Students.Name);

printf("\nStudent ID: %s", Students.StudentID);

printf("\nMid 1 marks: %d", Students.Mid1);

printf("\nMid 2 marks: %d", Students.Mid2);

printf("\nQuiz marks: %d", Students.Quiz);

printf("\nFinal marks: %d", Students.Final);

printf("\nTotal marks: %d", Students.Total);

printf("\nGPA: %.1f", Students.GPA);

}

int accumalateEvaluations(int A, int B, int C, int D, int E){

return(A+B+C+D+E);

}

float getGPA(int A, int B, int C, int D, int E){

int T=accumalateEvaluations(A,B,C,D,E);

if(T>=86){

return (4.0);

}

else if(86>T && T>=82){

return (3.6);

}

else if(82>T && T>=78){

return (3.0);

}

else if(78>T && T>=74){

return (2.6);

}

else if(74>T && T>=70){

return (2.3);

}

else if(70>T && T>=66){

return (2.0);

}

else if(66>T && T>=62){

return (1.6);

}

else if(58>T && T>=50){

return (1.0);

}

else {

return (0);

}

}

main(){

int ID;

printf("Enter student ID number only: ");

scanf("%d", &ID);

int i,j,N=10;

int Mode;

struct details Students[N];

for(;;){

printf("\n\n1. Input data\n2. Print data\n3. Finding a specific individual\n4. Printing Class Average\n5. Exit\n");

printf("Enter mode: ");

scanf("%d", &Mode);

switch(Mode){

case 1:{

for(i=0;i<N;){

if(i==7){

printf("Enter your name: ");

scanf("%s", &Students[i].Name);

printf("Enter your Roll Number: ");

scanf("%s", &Students[i].StudentID);

printf("Enter your Mid 1 Marks out of 15: ");

scanf("%d", &Students[i].Mid1);

printf("Enter your Mid 2 Marks out of 15: ");

scanf("%d", &Students[i].Mid2);

printf("Enter your Quiz Marks out of 10: ");

scanf("%d", &Students[i].Quiz);

printf("Enter your Assignment Marks out of 10: ");

scanf("%d", &Students[i].Assignment);

printf("Enter your Final Marks out of 50: ");

scanf("%d", &Students[i].Final);

Students[i].Total=accumalateEvaluations(Students[i].Mid1, Students[i].Mid2, Students[i].Quiz, Students[i].Assignment, Students[i].Final);

Students[i].GPA=getGPA(Students[i].Mid1, Students[i].Mid2, Students[i].Quiz, Students[i].Assignment, Students[i].Final);

i++;

}

char TempID[20]={"\0"};

printf("\nEnter Roll No. of student %d: ",i+1);

scanf("%s",&TempID);

for(j=0;j<N;j++){

if(strcmp(TempID,Students[j].StudentID)==0){

printf("Roll Number already exist. Enter new roll number.");

break;

}

else{

strcpy(Students[i].StudentID,TempID);

printf("Student Name: ");

scanf("%s", &Students[i].Name);

printf("Mid 1 Marks out of 15: ");

scanf("%d", &Students[i].Mid1);

printf("Mid 2 Marks out of 15: ");

scanf("%d", &Students[i].Mid2);

printf("Quiz out of 10: ");

scanf("%d", &Students[i].Quiz);

printf("Assignment out of 10: ");

scanf("%d", &Students[i].Assignment);

printf("Final Marks out of 50: ");

scanf("%d", &Students[i].Final);

Students[i].Total=accumalateEvaluations(Students[i].Mid1, Students[i].Mid2, Students[i].Quiz, Students[i].Assignment, Students[i].Final);

Students[i].GPA=getGPA(Students[i].Mid1, Students[i].Mid2, Students[i].Quiz, Students[i].Assignment, Students[i].Final);

i++;

break;

}

}

}

break;

}

case 2:{

printf("\n");

printf("\n\tStudent ID\tName\t\tGPA");

for(i=0;i<N;i++){

printf("\n\t%s\t%s\t", Students[i].StudentID, Students[i].Name);

if(Students[i].GPA==0){

printf("Fail");

}

else{

printf("%.1f",Students[i].GPA);

}

}

break;

}

case 3:{

char Input[100];

printf("\nEnter name or roll no. of person you want to find in correct format: ");

scanf("%s", &Input);

int Found=0;

for(i=0;i<N;i++){

if( strcmp(Students[i].Name, Input)==0 || strcmp(Students[i].StudentID, Input)==0 ){

smartFind(Students[i]);

Found=1;

}

}

if(Found==0){

printf("\nSorry, ID doesnot exist.");

}

break;

}

case 4:{

int SumMid1=0, SumMid2=0, SumFinal=0, SumTotal=0;

for(i=0;i<N;i++){

SumMid1+=Students[i].Mid1;

SumMid2+=Students[i].Mid2;

SumFinal+=Students[i].Final;

SumTotal+=Students[i].Total;

}

float avgMid1=SumMid1/N;

float avgMid2=SumMid2/N;

float avgFinal=SumFinal/N;

float avgTotal=SumTotal;

printf("\nAverage marks of all in Mid 1: %.2f",avgMid1);

printf("\nAverage marks of all in Mid 2: %.2f",avgMid2);

printf("\nAverage marks of all in Final: %.2f",avgFinal);

printf("\nAverage total score of all: %.2f",avgTotal);

int IMax;

for(i=0;i<N;i++){

if(Students[i].Total>Students[IMax].Total){

IMax=i;

}

}

printf("\n\nStudent ID %s has the maximum score of %d in the whole class.", Students[IMax].StudentID, Students[IMax].Total);

printf("\n\nList of students who didn't pass the course:");

for(i=0;i<N;i++){

if(Students[i].GPA==0){

printf("\n%s\t%s", Students[i].Name, Students[i].StudentID);

}

}

break;

}

case 5:{

printf("\n\nThank you for using!");

return 0;

break;

}

}

}

}

