//LINGUAGEM C: LISTA COMPLEMENTAR 4

//1

#include <stdio.h>

#define TAM 100

/\*Prototipos das FunÃ§Ãµes\*/

int SOMA(int v[], int n);

float MEDIA(int v[], int n);

void OrdenarCresc(int v[], int n);

void OrdenarDecresc(int v[], int n);

int MAIOR(int v[], int n);

int MENOR(int v[], int n);

void BUSCAR(int v[], int n, int elemento);

int ContElement(int v[], int n, int elemento);

void indiceMaior(int v[], int n, int maior);

void indiceMenor(int v[], int n, int menor);

int qtMaior(int v[], int n, int maior);

int qtMenor(int v[], int n, int menor);

void qtVezes(int v[], int q);

void showArray(int v[], int n);

int main(){

char op;

int valores[TAM], n, s, maior, menor, indice, cont, qt, elemento, i;

float media;

printf("\nInforme a quantidade de Numeros:\n");

scanf("%d",&n);

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

printf("\nInforme o %d elemento:\n",i+1);

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

}

system("clear");

while(1){

printf("\na-Somar elementos do vetor\n");

printf("\nb-Media AritmÃ©tica dos elementos do vetor\n");

printf("\nc-Ordenar o vetor em ordem Crescente\n");

printf("\nd-Ordenar o vetor em ordem Decrescente\n");

printf("\ne-Maior valor no vetor\n");

printf("\nf-Menor valor no vetor\n");

printf("\ng-Buscar um elemento no vetor (imprimir os Ã­ndices do elemento no vetor)\n");

printf("\nh-Contar o nÃºmero de ocorrÃªncias de um elemento no vetor\n");

printf("\ni-Imprimir os Ã­ndices do maior valor no vetor\n");

printf("\nj-Imprimir os Ã­ndices do menor valor no vetor\n");

printf("\nk-Contar o nÃºmero de ocorrÃªncias do maior valor no vetor\n");

printf("\nl-Contar o nÃºmero de ocorrÃªncias do menor valor no vetor\n");

printf("\nm-Imprimir os Q primeiros menores elementos do vetor ordenado crescente\n"); //Q == numeros

printf("\nn-Imprimir os Q primeiros maiores elementos do vetor ordenado decrescente\n");

printf("\no-Sair\n");

printf("\nOpÃ§Ã£o:");

getchar();

op = getchar();

system("clear");

if(op == 'o'){

break;

}

switch(op){

case 'a':

s = SOMA(valores, n);

printf("\nA Soma dos valores do vetor: %d\n",s);

break;

case 'b':

media = MEDIA(valores, n);

printf("\nA Media dos valores do vetor: %.1f\n",media);

break;

case 'c':

OrdenarCresc(valores, n);

printf("\nO vetor Ordenado de Forma Crescente:");

showArray(valores, n);

break;

case 'd':

OrdenarDecresc(valores, n);

printf("\nO vetor Ordenado de Forma Decrescente:");

showArray(valores, n);

break;

case 'e':

maior = MAIOR(valores, n);

printf("\nO Maior valor do vetor: %d\n",maior);

break;

case 'f':

menor = MENOR(valores, n);

printf("\nO Menor valor do vetor: %d\n", menor);

break;

case 'g':

printf("\nInforme qual o valor que deseja buscar:\n");

scanf("%d",&elemento);

printf("\nResultado da pesquisa: ");

BUSCAR(valores, n, elemento);

break;

case 'h':

printf("\nInforme qual o elemento:\n");

scanf("%d",&elemento);

cont = ContElement(valores, n, elemento);

if(cont > 0){

printf("\nElemento ocorre %d vezes dentro do vetor.\n",cont);

}else {

printf("\nElemento nao ocorre nenhuma fez!\n");

}

break;

case 'i':

printf("\nOs indices do Maior valor: ");

indiceMaior(valores, n, maior);

break;

case 'j':

printf("\nOs indices do Menor valor: ");

indiceMenor(valores, n, menor);

break;

case 'k':

maior = MAIOR(valores, n);

cont = qtMaior(valores, n, maior);

printf("\nO Maior valor ocorre %d vezes!\n",cont);

break;

case 'l':

menor = MENOR(valores, n);

cont = qtMenor(valores, n, menor);

printf("\nO menor valor ocorre %d vezes!\n",cont);

break;

case 'm':

printf("\nInforme os Q numeros menores:\n");

scanf("%d",&qt);

OrdenarCresc(valores, n);

qtVezes(valores, qt);

break;

case 'n':

printf("\nInforme os Q numeros maiores:\n");

scanf("%d",&qt);

OrdenarDecresc(valores, n);

qtVezes(valores, qt);

break;

default:

printf("\nOpcao Invalida!\n");

break;

}

}

printf("\n\n");

return 0;

}

int SOMA(int v[], int n){

int i, s;

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

s += v[i];

}

return s;

}

float MEDIA(int v[], int n){

int i, s;

float media;

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

s += v[i];

}

media = (float)s / n;

return media;

}

void OrdenarCresc(int v[], int n){

int i, y, aux;

//OrdenaÃ§Ã£o Crescente

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

for(y = i+1; y < n; y++){

if(v[i] > v[y]){

aux = v[i];

v[i] = v[y];

v[y] = aux;

}

}

}

}

void OrdenarDecresc(int v[], int n){

int i, y, aux;

//OrdenaÃ§Ã£o Decrescente

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

for(y = i+1; y < n; y++){

if(v[i] < v[y]){

aux = v[i];

v[i] = v[y];

v[y] = aux;

}

}

}

}

int MAIOR(int v[], int n){

int i, maior;

maior = v[0];

for(i = 1; i < n; i++){

if(v[i] > maior){

maior = v[i];

}

}

return maior;

}

int MENOR(int v[], int n){

int i, menor;

menor = v[0];

for(i = 1; i < n; i++){

if(v[i] < menor){

menor = v[i];

}

}

return menor;

}

void BUSCAR(int v[], int n, int elemento){

int i, flag;

flag = 0;

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

if(v[i] == elemento){

flag = 1;

printf("%d ",i);

}

}

if(flag == 0){

printf("Valor nao encontrado!");

}

}

int ContElement(int v[], int n, int elemento){

int i, cont;

cont = 0;

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

if(v[i] == elemento){

cont++;

}

}

return cont;

}

void indiceMaior(int v[], int n, int maior){

int i;

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

if(v[i] == maior){

printf("%d ",i);

}

}

}

void indiceMenor(int v[], int n, int menor){

int i;

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

if(v[i] == menor){

printf("%d ",i);

}

}

}

int qtMaior(int v[], int n, int maior){

int i, cont = 0;

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

if(v[i] == maior){

cont++;

}

}

return cont;

}

int qtMenor(int v[], int n, int menor){

int i, cont = 0;

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

if(v[i] == menor){

cont++;

}

}

return cont;

}

void qtVezes(int v[], int q){

int i;

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

printf("%d ",v[i]);

}

}

void showArray(int v[], int n){

int i;

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

printf("\nValor %d",v[i]);

}

}

//2

/\* Programa que lÃª os elementos de um vetor e em seguida faz diversas operaÃ§Ãµes no vetor. \*/

#include <stdio.h>

#include <string.h>

#define TAM 100

//ProtÃ³tipos de funÃ§Ãµes

int contaCaracteres(char str[]);

int contaLetras(char str[]);

int contaMinusculas(char str[]);

int contaMaiusculas(char str[]);

int contOcorrencias(char str[],char c);

void converteMinusculas(char str[]);

void converteMaiusculas(char str[]);

void substituiCaractere(char str[],char c1,char c2);

void inverte(char str[]);

int conPalavras(char c[]);

int contaVogais(char v[]);

int contaConsoantes(char v[]);

//Programa principal

int main() {

//DeclaraÃ§Ã£o de variÃ¡veis

int i, j, n, cont;

char str[TAM]; //string

char op, c, c1, c2;

//Leitura da string

printf("\nEntre com uma string - no mÃ¡ximo %d caracteres: ",TAM-1);

gets(str);

//LaÃ§o de execuÃ§Ã£o

while(1) { //laÃ§o "infinito" que termina com um comando break

//Imprime menu para o usuÃ¡rio

printf("\n\n==================================================================\n");

printf("\nEscolha uma das opÃ§Ãµes abaixo: ");

printf("\na - Contar o nÃºmero de caracteres da string");

printf("\nb - Contar o nÃºmero de letras da string");

printf("\nc - Contar o nÃºmero de letras minÃºsculas da string");

printf("\nd - Contar o nÃºmero de letras maiÃºsculas da string");

printf("\ne - Contar o nÃºmero de ocorrÃªncia de um caractere na string");

printf("\nf - Converter a string para letras minÃºsculas");

printf("\ng - Converter a string para letras maiÃºsculas");

printf("\nh - Substituir um caractere na string por outro caractere");

printf("\ni - Inverter a string");

printf("\nj - Contar o nÃºmero de palavras da string");

printf("\nk - Contar o nÃºmero de vogais da string");

printf("\nl - Contar o nÃºmero de consoantes da string");

printf("\nm - Sair");

printf("\n\nOpÃ§Ã£o: ");

op = getchar(); //lÃª opÃ§Ã£o do usuÃ¡rio

getchar(); //limpa buffer do teclado

printf("\n==================================================================\n");

//Verifica se a opÃ§Ã£o foi de saÃ­da do programa

if(op=='m') //se a opÃ§Ã£o for 'm' sai do laÃ§o de execuÃ§Ã£o com o comando break

break;

//Switch para escolha da execuÃ§Ã£o da opÃ§Ã£o

switch(op) {

case 'a': cont = contaCaracteres(str);

printf("\nQuant. de caracteres: %d",cont);

break;

case 'b': cont = contaLetras(str);

printf("\nQuant. de letras: %d",cont);

break;

case 'c': cont = contaMinusculas(str);

printf("\nQuant. de letras minÃºsculas: %d",cont);

break;

case 'd': cont = contaMaiusculas(str);

printf("\nQuantidade de letras Maiusculas: %d",cont);

break;

case 'e': cont = contOcorrencias(str);

printf("\nEntre com o caractere para busca: ");

c=getchar();

getchar();

cont=contaOcorrencias(str,c);

printf("\n Quant. de caracteres \'%c\' na string \"s\": %d",c,str,cont);

break;

case 'f': converteMinusculas(str);

printf("\nString convertida para minÃºsculas: \n%s",str);

break;

case 'g': converteMaiusculas(str);

printf("\nString convertida para maiusculas: \n%s",str);

break;

case 'h': printf("\nEntre com o caractere a ser substituido:");

c1=getchar();

getchar();

printf("\nEntre com o novo caractere:");

c2= getchar();

getchar();

substituiCaractere(str,c1c2);

printf(("String com o caractere \'%c\' substituido por \'%c\' : \n"%s\"",c1,c2,str);

break;

case 'i': inverte(str)

printf("\nstring invertida:\n%s",str);

break;

case 'j': cont = contPalavras(str);

printf("\nQuant. de palavras na string \"%s\":%d",str,cont);

break;

case 'k': cont = contaVogais(str);

printf("\nQuant.de vogais na string \"%s\":%d",str,cont);

break;

case 'l': cont = contaConsoantes(str);

printf("\nQuant. de consoantes na string\"%s\":%d",str,cont);

break;

default: printf("\nOpÃ§Ã£o invÃ¡lida!");

} //fim do switch de opÃ§Ãµes

} //fim do laÃ§o de execuÃ§Ã£o

printf("\n\n");

return 0;

} //fim do programa principal

//Conta a quantidade de caracteres

int contaCaracteres(char str[]) {

int i, cont;

cont = 0;

for(i=0;str[i]!='\0';i++)

cont = cont+1;

return cont;

}

//Conta a quantidade de letras

int contaLetras(char str[]) {

int i, cont;

cont = 0;

for(i=0;str[i]!='\0';i++)

if( (str[i]>='a'&&str[i]<='z') || (str[i]>='A'&&str[i]<='Z') )

cont = cont+1;

return cont;

}

//Conta a quantidade de letras minÃºsculas

int contaMinusculas(char str[]) {

int i, cont;

cont = 0;

for(i=0;str[i]!='\0';i++)

if(str[i]>='a'&&str[i]<='z')

cont = cont+1;

return cont;

}

//Conta a quantidade de letras maiusculas

int contaMaiusculas(char str[]) {

int i, cont;

cont = 0;

for(i=0;str[i]!='\0';i++)

if(str[i]<='a'&&str[i]<='z')

cont = cont+1;

return cont;

}

//conta a quantidade de ocorrencia

int contaOcorrencias(char str[],char c){

int i, cont;

cont = 0;

for(i=0;str[i]!='\0';i++)

if(str[i]==c)

cont++;

return cont;

}

//converte para minusculas

void converteMinusculas(char str[]){

int i;

for(i=0;str[i]!='\0';i++)

if(str[i]>='A'&&str[i]<='Z')

str[i]=str[i]+32;

}

//converte para maiusculas

void converteMaiusculas(char str[]){

int i;

for(i=0;str[i]!='\0';i++)

if(str[i]>='a'&&str[i]<='z')

str[i]=str[i]-32;

}

//Substitui Caractere

void substituiCaractere(char str[],char c1,char c2){

int i;

for(i=0;str[:]!='\0';i++)

if(str[i]==c1)

str[i]=c2;

}

//Inverte caractere

void inverte(char str[]){

int i,n;

char aux;

n=strlen(str);

for(i=0;i<n/2;i++)

aux=str[i];

aux=str[i];

str[i]=str[n-1-i];

str[n-1-i]=aux;

}

}

//conta palavras

int contPalavras(char c[]){

int i,cont;

cont=0;

for(i=0;c[i]!='\0';i++){

if(c[i]==' '){

cont++;

}

}

return cont++;

}

//conta vogais

int contVogais(char v[]){

int i,cont;

cont=0;

for(i=0;v[i]!='\0';i++){

if(v[i]=='a'|| v[i]=='e'|| v[i]=='i'|| v[i]=='o'|| v[i]=='u'|| v[i]=='A'|| v[i]=='E'|| v[i]=='I'|| v[i]=='O'|| v[i]=='U')

cont++;

}

return cont;

}

//conta consoantes

int contaConsoantes(char v[]) {

int i,cont=0;

for(i=0;v[i]!='\0';i++) {

if((v[i]>='A' && v[i]<='Z') || (v[i]>='a' && v[i]<='z')) && (v[i]!='A' && v[i]!='E' && v[i]!='I' && v[i]!='O' && v[i]!='U' && (v[i]!='a' && v[i]!='e' && v[i]!='i' && v[i]!='o' && v[i]!='u'))

cont++;

return cont;

}