//LINGUAGEM C: LISTA 09

//1

#include <stdio.h>

#define TAM 100

int soma(int \*a,int \*b);

int main() {

int z,n,i;

int x[TAM],y[TAM];

printf("\nEntre com a quantidade de numeros a serem somados de no maximo %d: ",TAM);

scanf("%d",&n);

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

printf("\n\nEntre com o primeiro numero: ");

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

printf("Entre com o segundo numero: ");

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

z = soma(x,y);

printf("Soma dos numeros: %d",z);

}

printf("\n\n");

getch();

return 0;

}

int soma(int \*a,int \*b) {

return \*a+ \*b;

}

//2

#include <stdio.h>

#define TAM 100 /\* tamanho mÃ¡ximo do vetor de raios \*/

#define PI 3.14 /\* define valor para PI \*/

void area\_esfera(float \*r, float \*a, int n);

int main() {

int i, N;

float raio[TAM], area[TAM];

/\* Entrada de dados \*/

printf("\n\*\*\*\*\* Calcula a Ã¡rea de esferas \*\*\*\*\*");

printf("\n\nEntre com o nÃºmero de esferas - no mÃ¡ximo 100: ");

scanf("%d",&N);

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

{

printf("Entre com o raio em metros da esfera %d: ", i+1);

scanf("%f", &raio[i]);

}

/\* Calcula as Ã¡reas atravÃ©s da funÃ§Ã£o \*/

area\_esfera(raio,area,N);

/\* Imprime os resultados\*/

printf("\n\*\* Resultados \*\* ");

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

{

printf("\nÃrea da esfera %d em metros quadrados: %10.2f", i+1, area[i]);

}

printf("\n\n");

return 0;

}

/\* FunÃ§Ã£o que calcula a Ã¡rea de esferas \*/

void area\_esfera(float \*r, float \*a, int n)

{

/\* r - ponteiro para os raios \*/

/\* a - ponteiro para as Ã¡reas \*/

/\* n - quantidade de Ã¡reas a ser calculadas \*/

int i;

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

{

a[i] = 4\*PI\*r[i]\*r[i];

}

}

//3

//IMCOPLETA... SO O METODO...

int main() {

char str[5];

converte(str);

.

.

.

funÃ§Ã£o:

void converte(char \*str) {

int i;

//4

#include <stdio.h>

#include <string.h>

#define TAM 100

void troca\_caractere(char \*p, char a, char b);

int main() {

int i,n;

char s[TAM], a, b;

printf("\nEntre com a string de no maximo %d: ",TAM);

gets(s);

printf("\nEntre com caractere a ser trocado: ");

a=getchar();

getchar();

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

b=getchar();

troca\_caractere(s,a,b);

printf("\n\n");

getch();

return 0;

}

void troca\_caractere(char \*p, char a, char b) {

int i;

char nome[TAM];

strcpy(nome,p);

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

if(nome[i] == a) {

nome[i] = b;

}

}

printf("\nString Original: ");

puts(p);

printf("\nString Resultado: ");

puts(nome);

}

//7

#include <stdio.h>

#include <stdlib.h>

void converte\_temp(float \*tc, float \*tf, int N); /\* protÃ³tipo da funÃ§Ã£o de conversÃ£o \*/

float media(float \*v, int N); /\* protÃ³tipo da funÃ§Ã£o que calcula a mÃ©dia \*/

int main() {

int i, N;

float \*temp\_cel, \*temp\_fah;

float media\_cel, media\_fah;

/\* Limpa tela \*/

system("clear");

/\* Entrada de dados \*/

printf("\n\*\*\*\*\* Faz a mÃ©dia e conversÃ£o de N temperaturas em Celsius e em Fahrenheit \*\*\*\*\*");

printf("\n\nEntre com a quantidade de temperaturas em graus Celsius: ");

scanf("%d",&N);

/\* AlocaÃ§Ã£o dinÃ¢mica de memÃ³ria \*/

temp\_cel = (float \*) malloc( N\*sizeof(float) );

if( temp\_cel == NULL ) {

printf("\n\nERRO - MemÃ³ria insuficiente");

exit(1);

}

temp\_fah = (float \*) malloc( N\*sizeof(float) );

if( temp\_fah == NULL ) {

printf("\n\nERRO - MemÃ³ria insuficiente");

exit(1);

}

/\* LÃª as temperaturas em graus Celsius \*/

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

{

printf("Entre com a temperatura %d em graus Celsius: ", i+1);

/\*scanf("%f", &temp\_cel[i]);\*/

scanf("%f", temp\_cel+i);

}

/\* Faz a conversÃ£o de Celsius para Fahrenheit \*/

converte\_temp(temp\_cel,temp\_fah,N);

/\* Imprime as temperaturas em Celsius e em Fahrenheit \*/

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

{

/\*printf("\nTemperatura %d em graus Celsius : %10.2f", i+1, temp\_cel[i]);

printf("\nTemperatura %d em graus Fahrenheit: %10.2f", i+1, temp\_fah[i]);\*/

printf("\nTemperatura %d em graus Celsius : %10.2f", i+1, \*(temp\_cel+i));

printf("\nTemperatura %d em graus Fahrenheit: %10.2f", i+1, \*(temp\_fah+i));

}

/\* Calcula a mÃ©dia das temperaturas \*/

media\_cel = media(temp\_cel, N);

media\_fah = media(temp\_fah, N);

/\* Imprime a mÃ©dia das temperaturas \*/

printf("\n\nA mÃ©dia das temperaturas em graus Celsius Ã© : %10.2f", media\_cel);

printf("\nA mÃ©dia das temperaturas em graus Fahreiheit Ã©: %10.2f", media\_fah);

/\* Libera memÃ³ria \*/

free(temp\_cel);

free(temp\_fah);

printf("\n\n");

return 0;

}

/\* FunÃ§Ã£o que calcula a conversÃ£o de graus Celsius para graus Fahrenheit \*/

void converte\_temp(float \*tc, float \*tf, int N)

{

/\* tc - Temperatura em graus Celsius \*/

/\* tf - Temperatura em graus Fahrenheit \*/

int i;

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

{

tf[i] = (tc[i]\*9.0)/5.0 + 32.0;

}

}

/\* FunÃ§Ã£o que calcula a mÃ©dia das temperaturas \*/

float media(float \*v, int N)

{

/\* v - ponteiro para as temperaturas \*/

/\* N - nÃºmero de temperaturas \*/

int i;

float med; /\* Armazena a mÃ©dia \*/

med = 0.0;

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

{

med = med + v[i];

}

med = med/N;

return med;

}

//8

#include <stdio.h>

#include <stdlib.h>

struct EFicha {

char nome[50];

long int telefone;

char logradouro[50];

int numero;

long int CEP;

char profissao[50];

};

int main() {

struct EFicha ficha;

/\* Limpa tela \*/

//system("clear");

/\* Entrada de dados \*/

printf("\n\*\*\*\*\* Faz o cadastro da ficha de um cliente \*\*\*\*\*\n");

printf("\nEntre com a ficha do cliente: ");

printf("\nNome: ");

gets(ficha.nome);

printf("Telefone: ");

scanf("%li",&ficha.telefone);

getchar();

printf("Logradouro: ");

gets(ficha.logradouro);

printf("Numero: ");

scanf("%li",&ficha.numero);

printf("CEP: ");

scanf("%li",&ficha.CEP);

getchar();

printf("Profissao: ");

gets(ficha.profissao);

/\* Imprime a ficha \*/

printf("\n\*\*\*\*\* Consulta de cadastro \*\*\*\*\*\n");

printf("\nFicha do cliente: ");

printf("\nNome: ");

puts(ficha.nome);

printf("Telefone: %li",ficha.telefone);

printf("\nLogradouro: ");

puts(ficha.logradouro);

printf("Numero: %li",ficha.numero);

printf("\nCEP: %li",ficha.CEP);

printf("\nProfissao: ");

puts(ficha.profissao);

printf("\n\n");

system("pause");

return 0;

}

//9

#include <stdio.h>

#include <stdlib.h>

struct EFicha {

char nome[50];

long int telefone;

char logradouro[50];

int numero;

long int CEP;

char profissao[50];

};

int main() {

int n,i,x;

struct EFicha ficha[50];

printf("\nEntre com a quantidade de fichas: ");

scanf("%d",&n);

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

printf("\n\*\*\*\*\* Faz o cadastro da ficha de um cliente \*\*\*\*\*\n");

printf("\nFicha do cliente %d: ",i+1);

printf("\nNome: ");

getchar();

gets(ficha[i].nome);

printf("Telefone: ");

scanf("%li",&ficha[i].telefone);

getchar();

printf("Logradouro: ");

gets(ficha[i].logradouro);

printf("Numero: ");

scanf("%li",&ficha[i].numero);

printf("CEP: ");

scanf("%li",&ficha[i].CEP);

getchar();

printf("Profissao: ");

gets(ficha[i].profissao);

}

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

printf("\n\*\*\*\*\* Consulta de cadastro \*\*\*\*\*\n");

printf("\nFicha do cliente %d: ",i+1);

printf("\nNome: ");

puts(ficha[i].nome);

printf("Telefone: %li",ficha[i].telefone);

printf("\nLogradouro: ");

puts(ficha[i].logradouro);

printf("Numero: %li",ficha[i].numero);

printf("\nCEP: %li",ficha[i].CEP);

printf("\nProfissao: ");

puts(ficha[i].profissao);

}

printf("\nEntre com o numero da inscricao da ficha: ");

scanf("%d",&x);

printf("\n\*\*\*\*\* Consulta de cadastro \*\*\*\*\*\n");

printf("\nFicha do cliente %d: ",x);

printf("\nNome: ");

puts(ficha[x-1].nome);

printf("Telefone: %li",ficha[x-1].telefone);

printf("\nLogradouro: ");

puts(ficha[x-1].logradouro);

printf("Numero: %li",ficha[x-1].numero);

printf("\nCEP: %li",ficha[x-1].CEP);

printf("\nProfissao: ");

puts(ficha[x-1].profissao);

printf("\n\n");

system("pause");

return 0;

}

//10

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct EFicha {

char nome[50];

long int telefone;

char logradouro[50];

long int numero;

long int CEP;

char profissao[50];

};

int main() {

char str[50];

int n,i,x;

struct EFicha \*ficha;

printf("\nEntre com a quantidade de fichas: ");

scanf("%d",&n);

//Alocacaoo dinamica de memoria

ficha = (struct EFicha\*)malloc(n\*sizeof(struct EFicha));

if(ficha==NULL) {

printf("\nERRO - MEMORIA INSUFICIENTE");

exit(1); //termina o programa

}

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

printf("\n\*\*\*\*\* Faz o cadastro da ficha de um cliente \*\*\*\*\*\n");

printf("\nFicha do cliente %d: ",i+1);

printf("\nNome: ");

getchar();

gets(ficha[i].nome);

printf("Telefone: ");

scanf("%ld",&ficha[i].telefone);

getchar();

printf("Logradouro: ");

gets(ficha[i].logradouro);

printf("Numero: ");

scanf("%ld",&ficha[i].numero);

printf("CEP: ");

scanf("%ld",&ficha[i].CEP);

getchar();

printf("Profissao: ");

gets(ficha[i].profissao);

}

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

printf("\n\*\*\*\*\* Consulta de cadastro \*\*\*\*\*\n");

printf("\nFicha do cliente %d: ",i+1);

printf("\nNome: ");

puts(ficha[i].nome);

printf("Telefone: %ld",ficha[i].telefone);

printf("\nLogradouro: ");

puts(ficha[i].logradouro);

printf("Numero: %ld",ficha[i].numero);

printf("\nCEP: %ld",ficha[i].CEP);

printf("\nProfissao: ");

puts(ficha[i].profissao);

}

printf("\nEntre com o nome do cliente: ");

gets(str);

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

if( strcmp(str,ficha[i].nome) == 0 ) {

printf("\n\*\*\*\*\* Consulta de cadastro \*\*\*\*\*\n");

printf("\nFicha do cliente %d: ",i+1);

printf("\nNome: ");

puts(ficha[i].nome);

printf("Telefone: %ld",ficha[i].telefone);

printf("\nLogradouro: ");

puts(ficha[i].logradouro);

printf("Numero: %ld",ficha[i].numero);

printf("\nCEP: %ld",ficha[i].CEP);

printf("\nProfissao: ");

puts(ficha[i].profissao);

}

}

free(ficha);

printf("\n\n");

getchar();

return 0;

}

//11

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct tipo\_endereco {

char rua[50];

int numero;

char bairro[30];

char cidade[50];

char sigla\_estado[3];

long int CEP;

};

struct ficha\_pessoal {

char nome[50];

long int telefone;

struct tipo\_endereco endereco;

};

struct ficha\_pessoal \*alocar\_memoria(int N); /\* protÃ³tipo de funÃ§Ã£o \*/

struct ficha\_pessoal \*liberar\_memoria(struct ficha\_pessoal \*p); /\* protÃ³tipo de funÃ§Ã£o \*/

void cadastro\_fichario(struct ficha\_pessoal \*p, int N); /\* protÃ³tipo de funÃ§Ã£o \*/

void consulta\_fichario(struct ficha\_pessoal \*p, int N); /\* protÃ³tipo de funÃ§Ã£o \*/

void consulta\_nome(struct ficha\_pessoal \*p, char \*nome, int N); /\* protÃ³tipo de funÃ§Ã£o \*/

int main() {

int i, N, op;

char nome[50];

struct ficha\_pessoal \*p;

/\* Limpa a tela \*/

system("clear");

/\* Entrada de nÃºmero de fichas \*/

printf("\nEntre com a quantidade de fichas: ");

scanf("%d",&N);

getchar();

/\* Aloca memÃ³ria para N posiÃ§Ãµes struct ficha\_pessoal \*/

p = alocar\_memoria(N);

if(p == NULL)

exit;

/\* Cadastra os N clientes no fichÃ¡rio \*/

cadastro\_fichario(p,N);

/\* Faz consulta completa ou por nome \*/

while(1) {

printf("\n\*\* Tecle a oÃ§Ã£o desejada \*\*");

printf("\n(1) - Consulta por nome");

printf("\n(2) - Consulta fichÃ¡rio completo");

printf("\nOpÃ§Ã£o: ");

scanf("%d",&op);

getchar();

switch (op) {

case 1: printf("\nEntre com um nome: ");

gets(nome);

consulta\_nome(p,nome,N);

break;

case 2: consulta\_fichario(p,N);

break;

}

printf("\nSe desejar continuar consulta tecle ENTER, senÃ£o tecle ESC: ");

scanf("%c",&op);

if (op != 27)

continue;

break;

}

/\* Libera memÃ³ria \*/

p = liberar\_memoria(p);

printf("\n\n");

}

/\* FunÃ§Ã£o que aloca a memÃ³ria com N posiÃ§Ãµes struct ficha\_pessoal \*/

struct ficha\_pessoal \*alocar\_memoria(int N)

{

struct ficha\_pessoal \*p;

/\* Verifica parÃ¢metro \*/

if(N<1) {

printf("\n\*\*\* Erro: parÃ¢metro invÃ¡lido \*\*\*\n");

return (NULL);

}

/\* Aloca a memÃ³ria com N posiÃ§Ãµes struct ficha\_pessoal \*/

p = (struct ficha\_pessoal \*) malloc( N\*sizeof(struct ficha\_pessoal) );

if(p == NULL) {

printf("\*\* Erro: MemÃ³ria Insuficiente \*\*");

return (NULL);

}

return (p); /\* retorna o ponteiro para a Ã¡rea de memÃ³ria alocada \*/

}

/\* FunÃ§Ã£o que libera a memÃ³ria \*/

struct ficha\_pessoal \*liberar\_memoria(struct ficha\_pessoal \*p)

{

if(p == NULL) return (NULL);

free(p); /\* Libera ponteiro \*/

return (NULL); /\* retorna ponteiro NULO \*/

}

/\* FunÃ§Ã£o que faz o fichÃ¡rio de N clientes \*/

void cadastro\_fichario(struct ficha\_pessoal \*p, int N)

{

int i;

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

printf("\nEntre com a ficha do cliente %d: ", i+1);

printf("\nNome: ");

gets(p[i].nome);

printf("Telefone: ");

scanf("%li",&p[i].telefone);

getchar();

printf("Rua: ");

gets(p[i].endereco.rua);

printf("NÃºmero: ");

scanf("%d",&p[i].endereco.numero);

getchar();

printf("Bairro: ");

gets(p[i].endereco.bairro);

printf("Cidade: ");

gets(p[i].endereco.cidade);

printf("Sigla Estado: ");

gets(p[i].endereco.sigla\_estado);

printf("CEP: ");

scanf("%li",&p[i].endereco.CEP);

getchar();

}

}

/\* FunÃ§Ã£o que faz a consulta do fichÃ¡rio completo \*/

void consulta\_fichario(struct ficha\_pessoal \*p, int N)

{

int i;

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

printf("\nFicha do cliente %d: ", i+1);

printf("\nNome: ");

puts(p[i].nome);

printf("Telefone: %li", p[i].telefone);

printf("\nRua: ");

puts(p[i].endereco.rua);

printf("NÃºmero: %d",p[i].endereco.numero);

printf("\nBairro: ");

puts(p[i].endereco.bairro);

printf("Cidade: ");

puts(p[i].endereco.cidade);

printf("Sigla Estado: ");

puts(p[i].endereco.sigla\_estado);

printf("CEP: %li\n",p[i].endereco.CEP);

}

}

/\* FunÃ§Ã£o que faz a consulta de uma ficha especÃ­fica pelo nome \*/

void consulta\_nome(struct ficha\_pessoal \*p, char \*nome, int N)

{

int i;

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

{

if( strcmp(nome,p[i].nome) == 0 )

{

printf("\nFicha do cliente %d: ", i+1);

printf("\nNome: ");

puts(p[i].nome);

printf("Telefone: %li", p[i].telefone);

printf("\nRua: ");

puts(p[i].endereco.rua);

printf("NÃºmero: %d",p[i].endereco.numero);

printf("\nBairro: ");

puts(p[i].endereco.bairro);

printf("Cidade: ");

puts(p[i].endereco.cidade);

printf("Sigla Estado: ");

puts(p[i].endereco.sigla\_estado);

printf("CEP: %li\n",p[i].endereco.CEP);

break;

}

}

}

//12

#include <stdio.h>

struct Eturma {

char nome[50];

float nota1;

float nota2;

float media;

};

int main() {

int n,i;

float mediaGeral=0;

struct Eturma aluno[100];

printf("\nEntre com o numero de alunos: ");

scanf("%d",&n);

getchar();

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

printf("\n\nNome do aluno %d: ",i+1);

gets(aluno[i].nome);

printf("Nota 1 do aluno %d: ",i+1);

scanf("%f",&aluno[i].nota1);

printf("Nota 2 do aluno %d: ",i+1);

scanf("%f",&aluno[i].nota2);

getchar();

aluno[i].media = (2\*aluno[i].nota1 + 3\*aluno[i].nota2)/5;

printf("Media Aritmetica do aluno %d: %.2f",i+1,aluno[i].media);

mediaGeral = mediaGeral + aluno[i].media;

if(aluno[i].media>=6)

printf("\nAluno Aprovado");

else

printf("\nAluno Reprovado");

}

mediaGeral = mediaGeral/n;

printf("\n\nMedia Geral: %.2f",mediaGeral);

printf("\n\n");

return 0;

}