leitura[strlen(leitura)-1] = '\0';

LPG

aula 14/08

#include <stdio.h>

int main(void){

int x=0;

int y=0;

int z=0;

scanf("%d %d %d", &x, &y, &z);

if(x+y <= z || y+z <= x || x+z <= y)

printf("Não forma um triangulo\n");

else

if (x == y && y == z)

printf("É um triang. Equilatero\n");

else

if(x == y || x == z || y == z)

printf("Isosceles\n");

else

printf("Escaleno \n");

}

quest 1

quadrante.c

#include <stdio.h>

int main(void){

float x=0, y=0;

scanf("%f %f", &x, &y);

if(x==0 && y==0) printf("Origem\n");

else if (x != 0 && y == 0) printf("Eixo Y\n");

else if (x == 0 && y != 0) printf("Eixo X\n");

else if ( x > 0 && y > 0) printf("Q1\n");

else if (x > 0 && y < 0) printf("Q4\n");

else if (x < 0 && y > 0) printf("Q2\n");

else if (x < 0 && y < 0) printf("Q3\n");

return 0;

}

produto.c

#include <stdio.h>

int main(void){

int codigo;

scanf("%d",&codigo);

switch(codigo){

case 1:

printf("Alimento perecivel\n");

break;

case 2 ... 4:

printf("Alimento nao perecivel\n");

break;

case 5 ... 6:

printf("Vestuario\n");

break;

case 7 ... 15:

printf("Higiene\n");

break;

default:

printf("Codigo invalido\n");

}

return 0;

}

quest 2

#include <stdio.h>

int main(void){

int a,b,c;

scanf("%d %d %d",&a ,&b ,&c);

if(a<b && a<c){

if(b<c){

printf("%d\n%d\n%d\n\n",a,b,c);

}else if(b>c){

printf("%d\n%d\n%d\n\n",a,c,b);

}

}else if(b<c && b<a){

if(a<c){

printf("%d\n%d\n%d\n\n",b,a,c);

}else if(a>c){

printf("%d\n%d\n%d\n\n",b,c,a);

}

}else{

if(a<b){

printf("%d\n%d\n%d\n\n",c,a,b);

}else if(a>b){

printf("%d\n%d\n%d\n\n",c,b,a);

}

}

printf("%d\n%d\n%d\n",a,b,c);

return 0;

}

quest 3.

#include <stdio.h>

int main(void) {

int x=0;

int y=0;

scanf("%d %d", &x, &y);

if(y%x == 0)

printf("São Múltiplos\n");

else

printf("Nao sao multiplos\n");

return 0;

}

quest 4

#include <stdio.h>

int main(void)

{

float x=0;

float y=0;

float z=0;

float a=0;

float b=0;

float c=0;

scanf("%f %f %f", &x, &y, &z);

if(x>=y && x>=z){

if(y>=z)

{

a=x;

b=y;

c=z;

}

else

{

a=x;

b=z;

c=y;

}

}

else if(y>=x && y>=z)

{

if(x>=z)

{

a=y;

b=x;

c=z;

}

else

{

a=y;

b=z;

c=x;

}

}

else

{

if(x>=y)

{

a=z;

b=x;

c=y;

}

else

{

a=z;

b=y;

c=x;

}

}

if(a>=b+c)

printf("NAO FORMA UM TRIANGULO\n");

else if((a\*a)==(b\*b)+(c\*c))

printf("TRIANGULO RETANGULO\n");

else if((a\*a)>(b\*b)+(c\*c))

printf("TRIANGULO OBTUSANGULO\n");

else if((a\*a)<(b\*b)+(c\*c))

printf("TRIANGULO ACUTANGULO\n");

if((a==b) || (b==c) || (a==c))

{

printf("????");

if(a==b && a==c)

printf("TRIANGULO EQUILATERO\n");

else

printf("TRIANGULO ISOSCELES\n");

}

return 0;

}

quest 5

#include <stdio.h>

int main(void) {

int x=0;

int y=0;

scanf("%d %d", &x, &y);

if(x==y)

printf("o jogo durou 24 hora(s)\n");

else if(x<y)

printf("o jogo durou %d hora(s)\n", y-x);

else

printf("o jogo durou %d hora(s)\n",24-(x-y));

return 0;

}

quest 6

#include <stdio.h>

int main (void){

int i=0;

for(i=1; i<=100; i++)

{

if(i%2 == 0)

printf("%d\n", i);

}

return 0;

}

quest 7

#include <stdio.h>

int main(void)

{

int n=0;

int cont\_pos=0;

int cont\_neg=0;

int cont\_impar=0;

int cont\_par=0;

int i=0;

while(i<5)

{

i++;

scanf("%d", &n);

if(n>0){

cont\_pos++;

if(n%2==0)

cont\_par++;

else

cont\_impar++;

}

else if(n<0){

cont\_neg++;

if(n%2==0)

cont\_par++;

else

cont\_impar++;

}

else if(n==0)

cont\_par++;

}

printf("%d valor(es) par(es)\n", cont\_par);

printf("%d valor(es) impar(es)\n", cont\_impar);

printf("%d valor(es) positivo(s)\n", cont\_pos);

printf("%d valor(es) negativo(s)\n", cont\_neg);

return 0;

}

quest 8

#include <stdio.h>

int main (void){

int i=0;

int j=0;

scanf("%d", &j);

for(i=1; i<11; i++)

{

printf("%d X %d = %d\n", i, j, i\*j);

}

return 0;

}

quest 9

#include <stdio.h>

int main(void)

{

int i=0;

int n=0;

float x=0;

float y=0;

float z=0;

scanf("%d", &n);

while(i<n)

{

scanf("%f %f %f", &x, &y, &z);

printf("%.1f\n", (x\*0.2)+(y\*0.3)+(z\*0.5));

i++;

}

return 0;

}

quest 10

#include <stdio.h>

int main(void){

int i, x, y, soma=0;

scanf("%d %d",&x,&y);

if(x>y){

for(i=(x-1);i>y;i--){

if(i%2!=0){

soma=soma+i;

}

}

}else{

for(i=(y-1);i>x;i--){

if(i%2!=0){

soma=soma+i;

}

}

}

printf("%d\n",soma);

return 0;

}

raiz.c

#include <stdio.h>

int main(void)

{

int n=0;

int i=1;

int cont=0;

scanf("%d", &n);

while(n>=i)

{

n=n-i;

cont++;

i+=2;

}

printf("%d\n", cont);

return 0;

}

/\*while(n>=i)

{

if(i%2!=0){

n=n-i;

cont++;

}

i++;

printf("%d\n", cont);

}

return 0;

}

Aula 21/08

Funções

fatorial.c

#include <stdio.h>

int fatorial(int x);

int main(void){

int n=0;

scanf("%d",&n);

int fat = fatorial(n);

printf("%d = %d", n, fat);

return 0;

}

int fatorial(int x)

{

int f=1;

int i=0;

for(i=1;i<=x;i++)

f\*=i;

return f;

}

/\* int fatorial(int x)

{

int f=1;

while(x>0)

{

f\*=x;

x--;

}

return f;

} \*/

Primo.c

#include <stdio.h>

int primo(int x);

int ver\_primo(int x);

int main(void)

{

int n=0;

scanf("%d", &n);

if(primo(n))

printf("primo\n");

else

printf("Não é primo\n");

return 0;

}

int primo(int x)

{

if(ver\_primo(x)==2)

return 1;

else

return 0;

}

int ver\_primo(int x)

{

int i=0;

int cont=0;

for(i=1;i<=x;i++)

{

if(x%i==0)

cont++;

}

return cont;

}

aula dia 23/08

#include <stdio.h>

float potencia(float base, int expo);

int main(void){

float base;

int expo;

scanf("%f %d",&base,&expo);

float p=potencia(base, expo);

printf("%.1f elevado a %d = %.1f\n",base,expo,p);

return 0;

}

float potencia(float base, int expo){

float pot=1;

while(expo>0){

pot\*=base;

expo--;

}

return pot;

}

conversão

#include <stdio.h>

int main(void){

int x, h, m, s, temp;

scanf("%d",&x);

h=x/3600;

temp=x%3600;

m=temp/60;

s=temp%60;

printf("%ds é equivalente a %dh %dm %ds\n",x,h,m,s);

return 0;

}

conversão com referencia

#include <stdio.h>

void converte(int x,int \*h,int \*m,int \*s);

int main(void){

int x, horas, min, seg;

scanf("%d",&x);

converte(x,&horas,&min,&seg);

printf("%ds é equivalente a %dh %dm %ds\n",x,horas,min,seg);

return 0;

}

void converte(int x,int \*h,int \*m,int \*s){

int temp;

\*h=x/3600;

temp=x%3600;

\*m=temp/60;

\*s=temp%60;

}

ponteiros

#include <stdio.h>

int main(void)

{

int a=3;

int b=4;

printf("A = %d\n",a);

printf("A = %p\n", &a);

int \*p1;

int \*p2;

p1 = &a;

printf("A = %d \n", \*p1);

\*p1=0; //a=0;

printf("end. A = %d\n",p1);

printf("%d\n", \*p1);

printf("%d \n", \*p2);

p2=&b;

\*p1=3;

int c=a\*b;//c=\*p1\*\*p2;

return 0;

}

aula dia 28/08

a.c

#include <stdio.h>

#include <math.h>

int calc\_raiz(float a, float b, float c, float \*p1, float \*p2);

int main(void)

{

float a=0;

float b=0;

float c=0;

float x1=0;

float x2=0;

scanf("%f %f %f", &a, &b, &c);

if(calc\_raiz(a,b,c,&x1,&x2))

printf("x1 = %.1f e x2 = %.1f\n", x1, x2);

else

printf("Nao ha raizes reais\n");

return 0;

}

int calc\_raiz(float a, float b, float c, float \*p1, float \*p2)

{

float delta = pow(b,2) -4\*a\*c;

if(delta<0)

return 0;

else{

\*p1 = (-b +sqrt(delta))/(2\*a);

\*p2 = (-b -sqrt(delta))/(2\*a);

return 1;

}

}

Recursividade

ex: fatorial:

x!{1, se x=0 ou x=1

{x(x-1)!, caso contrario

#include <stdio.h>

int fat\_rec(int x);

int main(void)

{

int n=0;

scanf("%d", &n);

int f=fat\_rec(n);

printf("%d! = %d\n", n, f);

return 0;

}

int fat\_rec(int x)

{

if(x==0 || x==1)

return 1;

else

return x\*fat\_rec(x-1);

}

potenciação

#include <stdio.h>

int pot\_rec(int base,int expo);

int main(void){

int base,expo;

scanf("%d %d",&base,&expo);

int f=pot\_rec(base,expo);

printf("%d^%d=%d\n",base,expo,f);

return 0;

}

int pot\_rec(int base,int expo){

if(expo==0)

return 1;

else

return base\*pot\_rec(base,expo-1);

}

aula dia 30/08

fibonacci

#include <stdio.h>

int main(void)

{

int i=1;

int a=1;

int b=1;

int n=0;

scanf("%d", &n);

printf("1\n1\n");

for(i=3;i<=n;i++)

{

int fim=a+b;

printf("%d\n", fim);

a=b;

b=fim;

}

return 0;

}

função

#include <stdio.h>

long long int fib(int n);

long long int fibo(int n);

int main(void)

{

int n=0;

int i=0;

int op=0;

scanf("%d", &n);

printf("1:recursivo\n2:iterativo\n");

scanf("%d", &op);

if(op==1)

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

printf("%d : %lld\n",i,fib(i));

else

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

printf("%d : %lld\n", i, fibo(i));

return 0;

}

long long int fib(int n){

if(n==1 || n==2)

return 1;

else

return fib(n-1)+fib(n-2);

}

long long int fibo(int n){

if(n==1 || n==2)

return 1;

else

{

long long int a=1;

long long int b=1;

long long int i=0;

long long int fim=0;

for(i=3;i<=n;i++)

{

fim=a+b;

a=b;

b=fim;

}

return fim;

}

}

hanoi

#include <stdio.h>

void hanoi(int n, char \*orig, char \*dest, char \*aux)

int main(void)

{

int n=0;

scanf("%d", &n);

hanoi(n,"origem","destino","aux");

}

void hanoi(int n, char \*orig, char \*dest, char \*aux){

if(n==1)

printf("move da %s p/ %s\n", orig, dest);

else{

hanoi(n-1,orig,aux,dest);

printf("move da %s p/ %s\n", orig, dest);

hanoi(n-1, aux, dest, orig);

}

}

aula dia 4/7

#include <stdio.h>

int main(void){

float v[10];

v[0]=1;

int i=0;

for(;i<10;i++){

printf("digite o %dº nº\n", i+1);

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

}

for(i=0;i<10;i++)

printf("%d : %.1f\n", i, v[i]);

}

printf("endereço de v: %p\n", v); //== &v;

return 0;

}

#include <stdio.h>

void le\_vetor(float v[10], int k); //v[] == \*v;

float media(float \*v,int k);

void mostra\_vetor(float \*v, int k);

int main(void){

float vet[10];

le\_vetor(vet,10);

float m=media(vet,10);

printf("Media = %.1f\n", m);

mostra\_vetor(vet,10);

}

void le\_vetor(float v[10], int k)

{

int i;

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

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

}

float media(float \*v,int k){

float soma=0;

int i;

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

soma += v[i];

}

return soma/k;

}

void mostra\_vetor(float \*v, int k){

int i=0;

for(;i<k;i++)

printf("vet [%d] = %.1f\n", i, v[i]);

}

#include <stdio.h>

int main(void){

int v[20],i=0,menor,i\_menor;

for(i=0;i<20;i++){

printf("Valor %d:\n",i);

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

if(i==0){

menor=v[i];

i\_menor=i;

}

if(v[i]<menor){

menor=v[i];

i\_menor=i;

}

}

printf("O menor valor é %d e sua posicao e %d\n",menor,i\_menor);

return 0;

}

dia 06/09

#include <stdio.h>

void mostra\_vetor(int \*v,int k);

void ordena(int \*v,int k);

int main(void){

int v[]={13,22,-1,7,18,22,30,0,-5,17};

mostra\_vetor(v,10);

ordena(v,10);

mostra\_vetor(v,10);

return 0;

}

void mostra\_vetor(int \*v, int k){

int i=0;

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

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

}

printf("\n");

}

void ordena(int \*p, int k){

int j=0;

for(j=0;j<k-1;j++){

int i,i\_menor=j;

for(i=j+1;i<k;i++){

if(p[i]<p[i\_menor])

i\_menor=i;

}

int aux = p[j];

p[j]=p[i\_menor];

p[i\_menor]=aux;

mostra\_vetor(p,k);

}

}

#include <stdio.h>

void mostra\_vetor(int \*v,int k);

int main(void){

int v[]={13,22,-1,7,18,22,30,0,-5,17};

int chave=0;

int i=0,cont=0;

printf("Digite o valor a ser bsucado: \n");

scanf("%d", &chave);

for(;i<10;i++)

{

if(v[i] == chave){

printf("Encontrado em %d\n", i);

cont++;

}

else if(cont!=0)

printf("Nao foi encontrado\n");

}

return 0;

}

dia 11/09 (world trade center)

busca

#include <stdio.h>

int busca(int \*p,int n,int chave);

int main(){

int v[]={3,-1,7,6,11,99,62,-4},x;

printf("Digite um valor:");

scanf("%d",&x);

int idx = busca(v,8,x);

if(idx==-1)

printf("Nao foi encontrado\n");

else

printf("Encontrado no indice %d\n",idx);

return 0;

}

int busca(int \*p,int n,int chave){

int i;

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

if(p[i]==chave)

return i;

return -1;

}

busca melhorada (divide pela metade a busca)

#include <stdio.h>

int busca(int \*p,int n,int chave);

int main(){

int v[]={2,3,6,8,10,11,12,18,22,31,34,37,42,51,66,71,88,89,90,99},x;

printf("Digite um valor:");

scanf("%d",&x);

int idx = busca(v,20,x);

if(idx==-1)

printf("Nao foi encontrado\n");

else

printf("Encontrado no indice %d\n",idx);

return 0;

}

int busca(int \*p,int n,int chave){

int ini=0,fim=n-1,meio;

do{

meio=(ini+fim)/2;

if(p[meio]==chave)

return meio;

else

if(p[meio]>chave)

fim=meio-1;

else

ini=meio+1;

}while(ini<=fim);

return -1;

}

busca recursiva

#include <stdio.h>

int bin(\*v, int n, int chave, int ini, int fim);

// int busca\_bin(int \*p, int n, int chave);

int main(void){

int v[]={2,3,6,8,10,11,12,18,22,31,34,37,42,51,66,71,88,89,90,99},x;

scanf("%d", &x);

int idx = bin(v,20,x,0,19);

if(idx == -1)

printf("Nao foi encontrado\n");

else

printf("encontrado no indice %d \n", idx);

return 0;

}

/\*int busca\_bin(int \*p, int n, int chave){

int ini=0,fim=n-1,meio;

do{

meio=(ini+fim)/2;

if(p[meio]==chave)

return meio;

else

if(p[meio]>chave)

fim=meio-1;

else

ini=meio+1;

}while(ini<=fim);

return -1;

}\*/

int bin(\*v, int n, int chave, int ini, int fim){

if(ini>fim)

return -1;

else{

int meio = (ini+fim)/2;

if(p[meio] == chave)

return meio;

else

if(p[meio]>chave)

return bin(p,n,chave,ini,meio-1);

else

return bin(p,n,chave,meio+1,fim);

}

}

aula 13/09

matrizes

#include <stdio.h>

int main(){

float mat[4][4];

int i,j,lin=4,col=4;

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

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

printf("[%d,%d]=",i+1,j+1);

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

}

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

for(j=0;j<col;j++)

printf("%4.1f",mat[i][j]);

printf("\n");

}

printf("Diagonal principal\n");

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

for(j=0;j<col;j++)

if(i==j)

printf("%.1f\n",mat[i][j]);

//printf("Diagonal principal\n");

//for(i=0;i<lin;i++)

// printf("%.1f\n",mat[i][i]);

printf("Diagonal secundaria\n");

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

for(j=0;j<col;j++)

if(i+j==lin-1)

printf("%.1f\n",mat[i][j]);

//printf("Diagonal secundaria\n");

//for(i=0;i<lin;i++)

// printf("%.1f\n",mat[i][lin-1-i]);

}

aula revisao

listas:

9.

#include <stdio.h>

int fat(int x);

int main(void){

int n=0, i=0;

scanf("%d", &n);

float s=0;

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

s+= (float) 1/(fat(i));

printf("S = %.1f\n", s);

}

return 0;

}

int fat(int x){

if(x==0)

return 1;

return x\*fat(x-1);

}

ou

#include <stdio.h>

int fat(int x);

int main(void){

int n=0, i=0;

scanf("%d", &n);

float s=1;

int den =1;

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

den\*=i;

s+=(float)1/den;

}

return 0;

}

int fat(int x){

if(x==0)

return 1;

return x\*fat(x-1);

}

versão final:

/////////////////////////////////////////////////

#include <stdio.h>

float serie(int i, int den, int n);

int main(void){

int n=0;

scanf("%d", &n);

float s=serie(1,1,n);

printf("S = %.1f\n", s);

return 0;

}

float serie(int i,int den, int n){

if(i<=n){

den\*=i;

return (float)1/den + serie(i+1,den,n);

}

else

return 1;

}

Prova 2

malloc

#include <stdio.h>

#include <stdlib.h>

int main(void){

float v[10];

int i;

for(i=0;i<10;i++){

printf("Digite o %d valor:",i+1);

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

}

for(i=0;i<10;i++)

printf("V[%d]=%.1f (end. %p)\n",i,v[i],&v[i]);

printf("End. %p\n",&v);

int n;

printf("Qual o tamanho do vetor?\n");

scanf("%d",&n);

float \*p = malloc(sizeof(float)\*n);

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

printf("P[%d]=",i);

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

}

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

printf("P[%d]=%.1f (end. %p)\n",i,p[i],&p[i]);

}

printf("Endereços de P: %p\n",&p);

free(p);

return 0;

}

aula 04/10

buscar numero

#include <stdio.h>

#include <stdlib.h>

int \*busca(int \*v,int k,int c){

int i,cont=0;

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

if(v[i]==c)

cont++;

int \*p=malloc(sizeof(int)\*(cont+1));

int j=0;

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

if(v[i]==c){

p[j]=i;

j++;

}

p[j]=-1;

return p;

}

/\*

int \*busca(int \*v,int k,int c){

int i,cont=0,\*p;

p=malloc(sizeof(int));

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

if(v[i]==c){

cont++;

p=realloc(p,sizeof(int)\*cont);

p[cont-1]=i;

}

p=realloc(p,sizeof(int)\*cont+1);

p[cont]=-1;

}

\*/

int main(void){

int v[]={3,8,7,-4,11,7,6,10,12,7,6};

int n=sizeof(v)/sizeof(int);

int chave;

printf("Digite um valor a ser buscado:");

scanf("%d",&chave);

int \*indices=busca(v,n,chave);

if(indices[0]==-1)//\*indices==-1

printf("Nao encontrado\n");

else{

int i;

printf("Encontrado em:\n");

for(i=0;indices[i]!=-1;i++)

printf("%d\n",indices[i]);

}

free(indices);

return 0;

}

aula 9/10

strcpy(var a, var b); -> copia var b pra var a

strcat(nome,str); -> concatenação;

int conta\_char(char \*s){

int cont=0;

while(s[cont]!='\0'){

cont++;

}

return cont;

}

void copia\_str(char \*dest,char \*orig){

int i;

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

dest[i]=orig[i];

}

dest[i]='\0';

}

void concatena\_str(char \*nome,char \*dasilva){

int i=0;

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

for(i, int j=0; dasilva[j] != '\0'; i++, j++)

nome[i] = dasilva[j];

}

uri

#include <stdio.h>

int main(void){

int t,i,j;

while(1){

scanf("%d", &t);

if(t==0)

return 0;

int m[t][t];

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

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

if(i==0 && j==0)

m[i][j]=1;

else if(j==0)

m[i][j]=m[i-1][j]\*2;

else

m[i][j]=m[i][j-1]\*2;

}

}

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

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

printf("%\*d\t",t,m[i][j]);

}

printf("\n");

}

}

return 0;

}

11/10

char nome[20];

char \*p;

p = nome; // p = &nome;

p=malloc(sizeof(char)\*30);

scanf("%s", p);

printf("%s\n", p);

free(p);

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

char\* aloca\_string();

int main(void){

printf("Digite um nome: \n");

char \*p = aloca\_string();

printf("Nome: %s\n", p);

free(p);

}

char\* aloca\_string(){

char buffer[101];

fgets(buffer,100,stdin);

char \*p = malloc(sizeof(char)\*(strlen(buffer)+1));

strcpy(p,buffer);

return p;

}

aula 16/10

#include <stdio.h>

int main(void){

int a=5;

int \*p1=&a;

int \*\*p2=&p1;

printf("A=%d\n",a);

printf("A(p1)=%d\n",\*p1);

printf("end. de A=%p\n",p1);

printf("end. de p1=%p\n",p2);

printf("end. de A=%p\n",\*p2);

printf("A(p2)=%d\n",\*\*p2);

}

#include <stdio.h>

#include <stdlib.h>

int main(void){

int lin,col,i,j;

scanf("%d %d",&lin,&col);

int \*\*p=malloc(sizeof(int\*)\*lin);

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

p[i]=malloc(sizeof(int)\*col);

}

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

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

printf("%d,%d:",i+1,j+1)

scanf("%d",&p[i][j])

}

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

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

printf("%4d ",p[i][j])

}

printf("\n");

}

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

free(p[i]);

}

free(p);

}

aula 18/10

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char \*aloca\_str(char\*msg){

char buffer[1001];

printf("%s",msg);

fflush(stdin);

fgets(buffer,1000,stdin);

char \*p=malloc(sizeof(char)\*strlen(buffer)+1);

strcpy(p,buffer);

return p;

}

int main(void){

int n,i;

scanf("%d",&n);

char \*\*nomes=malloc(sizeof(char\*)\*n);

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

nomes[i]=aloca\_str("Digite algo: ");

}

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

printf("%d:%s\n",i+1,nomes[i]);

}

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

free(nomes[i]);

}

free(nomes);

}

aula 23/10

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

int main(int argc,char \*argv[]){

if(argc!=3){

printf("Formato incorreto!\n");

printf("Formato esperado: potencia <base><expo>\n");

}else{

int base, expo;

sscanf(argv[1],"%d",&base);

sscanf(argv[2],"%d",&expo);

printf("Resultado:%.0lf\n", pow(base,expo));

}

return 0;

}

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char \*aloca\_str(char\*msg){

char buffer[1001];

printf("%s",msg);

fflush(stdin);

fgets(buffer,1000,stdin);

char \*p=malloc(sizeof(char)\*strlen(buffer)+1);

strcpy(p,buffer);

return p;

}

int main(void){

int n,i;

scanf("%d",&n);

char \*\*lista=malloc(sizeof(char\*)\*n);

le\_vetor(lista,n);

mostra\_vetor(lista,n);

char \*str=aloca\_str("Digite algo: ");

int i = busca(lista,n,str);

if(i==-1)

printf("Nao encontrado\n");

else

printf("%s encontrado em %d\n",str,i);

ordena(lista,n);

mostra(lista,n);

return 0;}

Lista 3 strings

1- #include <stdio.h>

int main(void){

char string[30];

int cont=0;

scanf("%s", string);

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

cont++;

printf("%d\n", cont);

return 0;

}

2- #include <stdio.h>

int main(void){

char string[30], letra;

int cont=0;

scanf("%s %c", string, &letra);

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

{

if(string[i] == letra)

cont++;

}

printf("%d\n", cont);

return 0;

}

3- #include <stdio.h>

int main(void){

char s1[30],s2[30],aux[60];

int i=0, j=0;

scanf("%s %s", s1, s2);

for(;s1[i] != '\0'; i++)

aux[i] = s1[i];

for(;s2[j] != '\0'; j++, i++)

aux[i] = s2[j];

printf("%s", aux);

return 0;

}

4- #include <stdio.h>

#include <string.h>

int main(void){

char s1[30];

int quanti=0, x=0,i=0;

scanf("%s", s1);

quanti = strlen(s1);

// printf("%d", quanti);

x=quanti;

for(i=0; i<quanti; i++, x--)

{

if(s1[i] != s1[x-1]){

printf("Nao e palindrome\n");

return 0;

}

}

printf("palindrome\n");

return 0;

}

ou

#include <stdio.h>

#include <string.h>

int main(void){

char s1[30];

scanf("%s",s1);

if(palindrome(s1))

printf("E palindrome\n");

else

printf("nao e\n");

}

int palindrome(char \*s){

char \*p1 = s;

char \*p2 = s+strlen(s)-1;

while(p1<p2){

if(\*p1 != \*p2)

return 0;

p1++;

p2--;

}

return 1;

}

5- #include <stdio.h>

#include <string.h>

int main(void){

char s1[30];

int quanti=0, x=0,i=0;

scanf("%s", s1);

quanti = strlen(s1);

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

{

if(i==quanti-1)

printf("%c\n", s1[i]-32);

else

printf("%c", s1[i]-32);

}

return 0;

}

6- #include <stdio.h>

#include <string.h>

void verifica(char \*s1, char letra, int quanti);

int main(void){

char s1[30], letra;

int quanti=0;

scanf("%s %c", s1, &letra);

quanti = strlen(s1);

verifica(s1,letra,quanti);

printf("%s\n", s1);

return 0;

}

void verifica(char \*s1, char letra, int quanti){

for(int i=0; i<quanti; i++)

{

if(s1[i] == letra)

s1[i] = ' ';

}

}

7- #include <stdio.h>

#include <string.h>

#include <stdlib.h>

char \*verifica(char \*s1, int quanti);

int busca(char \*v, int n, char c);

int main(void){

char s1[30], letra;

int quanti=0;

scanf("%s", s1);

quanti = strlen(s1);

char \*repetidas = verifica(s1,quanti);

printf("%s\n", repetidas);

return 0;

}

char \*verifica(char \*s1, int quanti){

char \*vet = malloc(sizeof(char)\*30);

int cont = 0, tot = 0;

for(int i=0; i<quanti; i++)

{

if(busca(vet, cont, s1[i]) == -1){

for(int j=i+1;j<quanti;j++)

{

if(s1[i]==s1[j])

{

vet[cont] = s1[j];

cont++;

break;

}

}

}

}

return vet;

}

int busca(char \*v, int n, char c){

int i;

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

if(v[i] == c)

return i;

return -1;

}

outra lista

a) p+i = 44; ==

b) p[i+2] = 900; \*(p+i+2)

c) \*(p+4) = 400; == p[4]

d) \*(p+2) + 4 = 816; == [p+2] + 4\*4;

e) p[i+1][3] = 6;

f) \*(p[i]+1) = 1; end. 304

g) &p[i+1] = 52;

h) \*(\*(p+4)+i) = 6;

i) p[i][i] = 3;

j) &p[i+2][i+1] = 916;

lista 4 quest 7

int \*\*c;

if(multiplica(a,lin\_a,col\_a,b,lin\_b,col\_b,&c,&lin\_c,&col\_c))

multiplica(int \*\*a,int lin\_a,int col\_a,int \*\*b,int lin\_b,int col\_b,int \*\*\*pc,int \*plc,int \*pcc){

if(col\_a != lin\_b){

return 0;

}

\*plc=lin\_a;

\*pcc=col\_b;

\*pc=malloc(sizeof(int\*)\*\*plc);

int i;

for(i=0;i<\*plc;i++){

(\*pc)[i]=malloc(sizeof(int)\*\*pcc);

}

//faz a multiplicacao

return 1;

}

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int verifica(char \*date);

int recursiva(char \*date);

int dia(char \*date, int a, int b);

int ano(char \*date, int a, int b, int c, int d);

int converte(char x){

return x - 48;

}

int main(void){

char date[11];

scanf("%s",date);

int ver = verifica(date);

if(ver){

int diaa = dia(date,0,1);

int mess = dia(date,3,4);

int anoo = ano(date,6,7,8,9);

printf("ta certo \n%d %d %d\n", diaa, mess, anoo);

}

else

printf("Nao ta certo\n");

// printf("%s\n",date);

return 0;

}

int verifica(char \*date){

if(date[2] != '/' || date[5] != '/')

return 0;

else

return recursiva(date);

}

int recursiva(char \*date){

int i;

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

{

if(i!=2 && i!= 5)

{

if(date[i] < '0' || date[i] > '9')

{

printf("%c %d\n", date[i], i);

return 0;

}

}

}

return 1;

}

int dia(char \*date, int a, int b){

int k1 = converte(date[a])\*10+converte(date[b]);

return k1;

}

int ano(char \*date, int a, int b, int c, int d){

int k1 = 1000\*converte(date[a])+100\*converte(date[b])+10\*date[c]+converte(date[d]);

return k1;

}

#include <stdio.h>

// #include <bits/stdc++.h>

struct ponto\_2D {

float x, y;

};

void mostra\_ponto(struct ponto\_2D p);

void le\_ponto (struct ponto\_2D \*p);

int main(void){

struct ponto\_2D p1, p2 = {1, 3};

p1.x = p1.y = 0;

printf("Coordenadas de P1:\n");

printf("x = %.1f\ny = %.1f\n", p1.x, p1.y);

printf("Coordenadas de P2:\n");

printf("x = %.1f\ny = %.1f\n", p2.x, p2.y);

p1=p2;

p1.x = p2.x;//copia somente x;

scanf("%f %f", &p2.x, &p2.y);

struct ponto\_2D v[10];

int i;

for(i = 0; i < 10; i++){

le\_ponto(v[i]);

}

for(i = 0; i < 10; i++){

mostra\_ponto(v[i]);

}

return 0;

}

void mostra\_ponto(struct ponto\_2D p){

printf("[%.1f, %.1f]\n", p.x, p.y);

}

void le\_ponto (struct ponto\_2D \*p) {

printf("coordenadas: ");

scanf("%f %f", &p->x, &p->y);

}

#include <stdio.h>

#include <stdlib.h>

typedef struct{

float x, y;

} ponto;

typedef struct{

ponto p1,p2;

} retangulo;

typedef struct{

ponto centro;

float raio;

} circulo;

int main(void){

ponto p2;

retangulo r1;

r1.p1.x = 0;

r1.p2 = p2;

}

aula 13/11

struct

#include <stdio.h>

#include <stdlib.h>

typedef struct{

int dia,mes,ano;

}Data;

typedef struct{

char descricao[20];

int quantidade;

float preco;

}Item;

typedef struct{

int codigo,quantidade;

Data data\_pedido;

float total;

Item \*itens;

}Pedido;

int main(void){

Pedido p;

printf("Digite o codigo: ");

scanf("%d",&p.codigo);

printf("Digite a data: ");

scanf("%d %d %d",&p.data\_pedido.dia,&p.data\_pedido.mes,&p.data\_pedido.ano);

printf("Digite a qtd de itens: ");

scanf("%d",&p.quantidade);

p.itens = malloc(sizeof(Item)\*p.quantidade);

int i;

p.total = 0;

for(i=0;i<p.quantidade;i++){

printf("\nDigite a descricao: ");

fgets(p.itens[i].descricao,19,stdin);

printf("Digite o preco unitario: ");

scanf("%f",&p.itens[i].preco);

printf("Digite a quantidade: ");

scanf("%d",&p.itens[i].quantidade);

p.total += p.itens[i].preco \* p.itens[i].quantidade;

}

//mostrar pedido na tela

printf("Codigo: %d\n",p.codigo);

printf("Data: %d/%d/%d\n",p.data\_pedido.dia,p.data\_pedido.mes,p.data\_pedido.ano);

printf("Qtd de itens: %d\n",p.quantidade);

for(i=0;i<p.quantidade;i++){

printf("\nDescricao: %s\n",p.itens[i].descricao);

printf("Preco unitario: %.2f\n",p.itens[i].preco);

printf("Quantidade: %d\n",p.itens[i].quantidade);

}

printf("Total: %.2f\n",p.total);

//desalocar o vetor itens

free(p.itens);

}

#include <stdio.h>

#include <stdlib.h>

typedef struct{

float x,y;

}Ponto\_2D;

typedef struct{

Ponto\_2D centro;

float raio;

}Circulo;

void le\_ponto(Ponto\_2D \*p);

void mostra\_ponto(Ponto\_2D p);

void le\_circulo(Circulo \*p);

void mostra\_circulo(Circulo x);

int main(void){

Ponto\_2D a, b;

le\_ponto(&a);

le\_ponto(&b);

mostra\_ponto(a);

mostra\_ponto(b);

Circulo c;

le\_circulo(&c);

mostra\_circulo(c);

}

void le\_ponto(Ponto\_2D \*p){

printf("Digite as coord. x e y:\n");

scanf("%f %f",&p->x,&p->y);

}

void mostra\_ponto(Ponto\_2D p){

printf("X=%.1f, Y=%.1f\n",p.x,p.y);

}

void le\_circulo(Circulo \*p){

le\_ponto(&p->centro);

printf("Digite o raio:\n");

scanf("%f",&p->raio);

}

void mostra\_circulo(Circulo x){

printf("Circulo de raio %.1f e centro em",x.raio);

mostra\_ponto(x.centro);

}

#include <stdio.h>

int main(void){

FILE \*in, \*out;

in = fopen("b.txt", "rt");

if(in==NULL){

printf("Erro no arquivo de entrada\n");

return -1;

}

out=fopen("saida.txt","wt");

if(out==NULL){

printf("Erro no arq de saida\n");

return -1;

}

char ch=getc(in);

while(ch!=EOF){

printf("%c",ch);

putc(ch,out);

ch = getc(in);

}

fclose(in);

fclose(out);

}

#include <stdio.h>

int main(void){

FILE \*in, \*out;

in = fopen("b.txt", "rt");

if(in==NULL){

printf("Erro no arquivo de entrada\n");

return -1;

}

out=fopen("saida.txt","wt");

if(out==NULL){

printf("Erro no arq de saida\n");

return -1;

}

char ch;

while(ch=getc(in)!=EOF){

printf("%c",ch);

putc(ch,out);

}

fclose(in);

fclose(out);

}

#include <stdio.h>

int main(void){

FILE \*in, \*out;

in = fopen("b.txt", "rt");

if(in==NULL){

printf("Erro no arquivo de entrada\n");

return -1;

}

out=fopen("saida.txt","wt");

if(out==NULL){

printf("Erro no arq de saida\n");

return -1;

}

char str[1001];

while(fgets(str,1000,in) != NULL){

printf("%s", str);

fputs(str,out);

}

fclose(in);

fclose(out);

}

produtor.c

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

int main(void){

char str[20];

int v[10],i;

scanf("%s",str);

for(i=0;i<10;i++)

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

FILE \*out=fopen("dados.bin","wb");

if(out==NULL){

printf("Erro no arquivo!\n");

return -1;

}

fwrite(str,sizeof(char),20,out);

fwrite(v,sizeof(int),10,out);

fclose(out);

return 0;

}

consumidor.c

#include <string.h>

#include <stdio.h>

#include <stdlib.h>

int main(void){

char s[20];

int v[10];

FILE \*in=fopen("dados.bin","rb");

if(in==NULL){

printf("Erro\n");

return -1;

}

fread(s,sizeof(char),20,in);

fread(v,sizeof(int),10,in);

int i;

for(i=0;i<10;i++)

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

fclose(in);

}