**Exercícios Introdução à Programação**

**Ficha 5**

1.

float quadrado(float num)

{

float quadrado;

quadrado=num\*num;

return quadrado;

}

int main(void)

{

float num;

do

{

printf("Introduza um número real:\n");

scanf("%f",&num);

if(num!=0)

{

printf("R: %.2f\n",quadrado(num));

}

}while(num!=0);

return 0;

}

2.

int entrelimites(int num,int liminf,int limsup)

{

int devolve=0;

if(liminf<limsup)

{

if(num>=liminf&&num<=limsup)

{

devolve=1;

}else if(num<liminf||num>limsup)

{

devolve=0;

}

return devolve;

}else if(liminf>=limsup)

{

return 0;

}

}

int main(void)

{

int num=0,liminf=0,limsup=0;

printf("Introduza o limite inferior:\n");

scanf("%d",&liminf);

printf("Introduza o limite superior:\n");

scanf("%d",&limsup);

printf("Introduza um número inteiro:\n");

scanf("%d",&num);

printf("%d",entrelimites(num,liminf,limsup));

}

3.

void linha(int n)

{

int i=1;

char ch='\*';

while(i<=n)

{

printf("%c ",ch);

i+=1;

}

}

int main(void)

{

int n;

printf("Introduza um número inteiro positivo:\n");

scanf("%d",&n);

linha(n);

}

4.

void quadrado(int lado)

{

int i=1,z=1;

char ch='\*';

while(z<=lado)

{

while(i<=lado)

{

printf("%c ",ch);

i+=1;

}

printf("\n");

z+=1;

i=1;

}

}

int main(void)

{

int lado;

printf("Introduza o lado do quadrado:\n");

scanf("%d",&lado);

quadrado(lado);

}

5. **Triângulo**

void triangulo(int altura)

{

int i,j;

char ch='\*';

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

{

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

{

printf("\* ");

}

printf("\n");

}

}

int main(void)

{

int altura;

printf("Introduza a altura do triângulo:\n");

scanf("%d",&altura);

triangulo(altura);

}

5. **Losango**

void losango(int altura)

{

int linha,space,symbol;

for(linha=1;linha<=altura;linha++)

{

for(space=altura-linha;space>=1;space--)

{

printf(" ");

}

for(symbol=1;symbol<=(linha\*2)-1;symbol++)

{

printf("\*");

}

printf("\n");

}

for(linha=altura-1;linha>=1;linha--)

{

for (space=1;space<=altura-linha;space++)

{

printf(" ");

}

for(symbol=1;symbol<=2\*linha-1;symbol++)

{

printf("\*");

}

printf("\n");

}

}

int main(void)

{

int altura;

printf("Introduza a altura do losango.\n");

scanf("%d",&altura);

losango(altura);

return 0;

}

6.

int triangulo(int linhasup)

{

int symbolsup=1,altura,spacext,spaceint,spacelinhainf=1,linha,contasymbol=linhasup;

altura=(linhasup-1)/2;

while(symbolsup<=linhasup)

{

printf("\*");

symbolsup++;

}

printf("\n");

for(linha=1;linha<=altura-1;linha+=1)

{

for(spacext=1;spacext<=linha;spacext++)

{

printf(" ");

}

printf("\*");

for(spaceint=1;spaceint<=linhasup-4;spaceint++)

{

printf(" ");

}

printf("\*");

printf("\n");

contasymbol+=2;

linhasup-=2;

}

while(spacelinhainf<=altura)

{

printf(" ");

spacelinhainf++;

}

printf("\*");

printf("\n");

contasymbol+=1;

return contasymbol;

}

int main(void)

{

int linhasup,res=0;

printf("Introduza o valor da linha superior [5;11]: ");

scanf("%d",&linhasup);

if(linhasup>=5&&linhasup<=11)

{

res=triangulo(linhasup);

printf("R: %d asteriscos.",res);

}

else

printf("O valor introduzido é inválido.");

}

7.

int trianguloinvert(int numlinhas)

{

int spaces,linha,numesq,numdir;

for(linha=1;linha<=numlinhas;linha++)

{

for(spaces=1;spaces<=linha-1;spaces++)

{

printf(" ");

}

for(numesq=1;numesq<=(numlinhas+1)-linha;numesq++)

{

printf("%d",numesq);

}

for(numdir=numlinhas-linha;numdir>=1;numdir--)

{

printf("%d",numdir);

}

printf("\n");

}

return 0;

}

int main(void)

{

int numlinhas;

printf("Introduza o número de linhas do triângulo [1;9] :");

scanf("%d",&numlinhas);

trianguloinvert(numlinhas);

return 0;

}

8.

int contadig(int num)

{

int contadigitos=0;

while(num!=0)

{

num=num/10;

contadigitos+=1;

}

return contadigitos;

}

int main(void)

{

int num;

printf("Introduza um número inteiro positivo:\n");

scanf("%d",&num);

if(num>0)

{

printf("O número introduzido tem %d dígitos.",contadig(num));

}else

{

printf("O número introduzido é inválido.\n");

}

return 0;

}

9.

int aproxima (float n,float m)

{

float media=(n+m)/2;

int aprox=0;

while(aprox!=(int)media)

{

aprox+=1;

}

return aprox;

}

int main(void)

{

float n,m;

printf("Introduza um número real:\n");

scanf("%f",&n);

printf("Introduza um número real:\n");

scanf("%f",&m);

printf("O número inteiro que mais se aproxima da média dos valores introduzidos é %d.", aproxima(n,m));

}

10.

int cubo(int numcopy)

{

int cube;

cube=numcopy\*numcopy\*numcopy;

return cube;

}

int somacubo (int num)

{

int somacube=0,numcopy,numY;

numcopy=num;

numY=num;

while(num>0)

{

numcopy=num%10;

somacube=somacube+cubo(numcopy);

num=num/10;

}

if(somacube==numY)

{

return 1;

}else{return 0;}

}

int main()

{

char ch;

int num;

do

{

do

{

printf("Introduza um número inteiro [100;999]:\n");

scanf("%d",&num);

}while (num<100||num>999);

if(somacubo(num)==1)

{

printf("O número introduzido satisfaz a condição.\n");

}else if(somacubo(num)==0)

{

printf("O número introduzido não satisfaz a condição.\n");

}

fflush(stdin);

printf("Pretende Introduzir outro número?\n");

scanf("%c",&ch);

}while (ch=='y'||ch=='Y');

return 0;

}

11.

int max(int dimensao)

{

int i=1,valormax=0,vezesmax=0,num;

while(i<=dimensao)

{

printf("Introduza um número:\n");

scanf("%d",&num);

if(num>valormax)

{

vezesmax=0;

valormax=num;

vezesmax+=1;

}else if(num==valormax)

{

vezesmax+=1;

}

i+=1;

}

return vezesmax;

}

int main()

{

int dimensao,maxvezes=0;

printf("Introduza a dimensão da sequência:\n");

scanf("%d",&dimensao);

maxvezes=max(dimensao);

printf("O número de vezes que o valor máximo surgiu foi %d.",maxvezes);

return 0;

}

12.

int minmult (int numA,int numB)

{

int minmultcomum,numAcopy,numBcopy;

numAcopy=numA;

numBcopy=numB;

do{

if(numA<numB)

{

numA+=numAcopy;

}

else if(numA>numB)

{

numB=numB+numBcopy;

}

}while(numA!=numB);

minmultcomum=numA;

return minmultcomum;

}

int main(void)

{

int numA=0,numB=0,mult=0;

printf("Introduza um número inteiro positivo:\n");

scanf("%d",&numA);

printf("Introduza um número inteiro positivo:\n");

scanf("%d",&numB);

mult=minmult(numA,numB);

printf("\nO mínimo múltiplo comum é: %d",mult);

}

14.

int ndiv (int num)

{

int ndivisores=0,i=1;

while(i<=num)

{

if((num%i)==0)

{

ndivisores+=1;

}

i+=1;

}

return ndivisores;

}

int main()

{

int num,numdiv=0;

do{

printf("Introduza um número inteiro positivo:\n");

scanf("%d",&num);

if(num>0)

{

numdiv=ndiv(num);

if(numdiv==2)

{

printf("Introduziu um número primo!\n");

}

}

}while(num>0);

}

16.

int verificavogal (char ch)

{

switch(ch)

{

case 'a': return 1;

break;

case 'A': return 1;

break;

case 'e': return 2;

break;

case 'E': return 2;

break;

case 'i': return 3;

break;

case 'I': return 3;

break;

case 'o': return 4;

break;

case 'O': return 4;

break;

case 'u': return 5;

break;

case 'U': return 5;

break;

default:return 0;

}

}

void main(void)

{

char ch;

int res;

printf("Introduza um caracter:\n");

scanf("%c",&ch);

res=verificavogal(ch);

if(res>=1&&res<=5)

{

printf("O caracter introduzido é uma vogal de ordem %d",res);

}

else if(res==0)

{

printf("O caracter introduzido não é uma vogal.\n");

}

}

18.

void conversor(int liminf,int limsup)

{

int num;

char ch;

num=liminf;

ch=num;

printf("\n\nCódigo Carácter\n");

while(num<=limsup)

{

printf(" %d %c\n",num,ch);

num+=1;

ch+=1;

}

}

int main(void)

{

int liminf,limsup;

printf("Introduza o limite inferior.\n");

scanf("%d",&liminf);

printf("Introduza o limite superior.\n");

scanf("%d",&limsup);

conversor(liminf,limsup);

return 0;

}

19.(a)

int fatorial (int num)

{

int i,factor=1;

i=num;

while(i>=1)

{

factor=factor\*i;

i-=1;

}

return factor;

}

19.(b)

long fatorialN (int N)

{

int i;

long factor=1;

i=N;

while(i>=1)

{

factor=factor\*(long)i;

i-=1;

}

return factor;

}

long fatorialP (int P)

{

int i;

long factor=1;

i=P;

while(i>=1)

{

factor=factor\*(long)i;

i-=1;

}

return factor;

}

long fatorialnmenosp (int nmenosp)

{

int i;

long factor=1;

i=nmenosp;

while(i>=1)

{

factor=factor\*(long)i;

i-=1;

}

return factor;

}

int main(void)

{

long expressao=0,condicao=0;

int N=0,P=0,nmenosp=0;

printf("Introduza o valor de N:\n");

scanf("%d",&N);

printf("Introduza o valor de P:\n");

scanf("%d",&P);

nmenosp=N-P;

if(N>=0&&P>=0)

{

expressao=(fatorialN(N))/((fatorialP(P))\*(fatorialnmenosp(nmenosp)));

printf("R: %ld",expressao);

}else {printf("Equação Impossível.");}

return 0;

}