Ex: 001

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include "pilha.h"

char \*posfixa(char \*e) {

   static char s[256];

   int j = 0;

   Pilha P = pilha(256);

   for(int i=0; e[i]; i++)

      if( isdigit(e[i]) ) s[j++] = e[i];

      else if( strchr("+\*-/",e[i]) ) empilha(e[i],P);

      else if( e[i]==')' ) s[j++] = desempilha(P);

   s[j] = '\0';

   destroip(&P);

   return s;

}

int main(void) {

   char e[513];

   printf("Infixa? ");

   gets(e);

   printf("Posfixa: %s\n", posfixa(e));

   return 0;

}

/\*

EX001:   a) 23\*84/+

         b) 95-1-

         c) 234\*5-

         d) 34+86-\*2/

\*/

Ex: 002

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include "pilha.h"

int prio(char o) {

   switch( o ) {

      case '(': return 0;

      case '+':

      case '-': return 1;

      case '\*':

      case '/': return 2;

   }

   return -1; // operador inválido!

}

char \*posfixa(char \*e) {

   static char s[256];

   int j = 0;

   Pilha P = pilha(256);

   for(int i=0; e[i]; i++)

      if( e[i]=='(' ) empilha('(',P);

      else if( isdigit(e[i]) ) s[j++] = e[i];

      else if( strchr("+-/\*",e[i]) ) {

         while( !vaziap(P) && prio(topo(P))>=prio(e[i]) )

            s[j++] = desempilha(P);

         empilha(e[i],P);

      }

      else if( e[i] == ')' ) {

         while( topo(P)!='(' )

           s[j++] = desempilha(P);

         desempilha(P);

      }

        while( !vaziap(P) )

            s[j++] = desempilha(P);

        s[j] = '\0';

        destroip(&P);

    return s;

}

int main(void) {

   char e[513];

   printf("Infixa? ");

   gets(e);

   printf("Posfixa: %s\n", posfixa(e));

   return 0;

}

/\*EX002: a) 23\*84/+

         b) 95-1-

         c) 234\*+5-

         d) 34+86-\*2/

\*/

Ex: 003

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include "pilha.h"

int valor(char \*e) {

   Pilha P = pilha(256);

   for(int i=0; e[i]; i++)

      if( isdigit(e[i]) ) empilha(e[i]-'0',P);

      else {

         int y = desempilha(P);

         int x = desempilha(P);

         switch( e[i] ) {

            case '+': empilha(x+y,P); break;

            case '-': empilha(x-y,P); break;

            case '\*': empilha(x\*y,P); break;

            case '/': empilha(x/y,P); break;

         }

   }

   int z = desempilha(P);

   destroip(&P);

   return z;

}

int prio(char o) {

   switch( o ) {

      case '(': return 0;

      case '+':

      case '-': return 1;

      case '\*':

      case '/': return 2;

   }

   return -1; // operador inválido!

}

char \*posfixa(char \*e) {

   static char s[256];

   int j = 0;

   Pilha P = pilha(256);

   for(int i=0; e[i]; i++)

      if( e[i]=='(' ) empilha('(',P);

      else if( isdigit(e[i]) ) s[j++] = e[i];

      else if( strchr("+-/\*",e[i]) ) {

         while( !vaziap(P) && prio(topo(P))>=prio(e[i]) )

            s[j++] = desempilha(P);

         empilha(e[i],P);

      }

      else if( e[i] == ')' ) {

         while( topo(P)!='(' )

           s[j++] = desempilha(P);

         desempilha(P);

      }

        while( !vaziap(P) )

            s[j++] = desempilha(P);

        s[j] = '\0';

        destroip(&P);

    return s;

}

int main(void) {

   char e[513];

   printf("Infixa? ");

   gets(e);

   printf("Posfixa: %s\n", posfixa(e));

   printf("Resultado: %d", valor(posfixa(e)));

   return 0;

}



Ex:004

/\*

   Com base nos algoritmos descritos, crie um programa para ler uma expressão booleana comple-tamente parentesiada e exibir sua forma posfixa correspondente, bem como seu valor numérico.

   Considere que as expressões são compostas por:

   Operandos: letras maiúsculas F e V, com valores numéricos 0 e 1, respectivamente.

   Operadores: ! (não), & (e) e | (ou), da maior para a menor prioridade.

   Delimitadores: parênteses de abertura e fechamento.

   Por exemplo, para a expressão booleana infixa parentesiada "((!F)|(F&V))", o programa deve apresentar como saída a forma posfixa "F!FV&|" e o valor numérico 1.

   (0 -> 1) ou (0 & 1 -> 0) -> 1

\*/

#include <stdio.h>

#include <ctype.h>

#include <string.h>

#include "pilha.h"

int Opou(char x, char y){

   if(x=='V')x=1; else x=0;

   if(y=='V')y=1; else y=0;

   if(x==1 || y==1 ){

      return 1;

   }

   else{

      return 0;

   }

   printf("%d",x);

   printf("%d\n",y);

}

int Ope(char x, char y){

   if(x=='V')x=1; else x=0;

   if(y=='V')y=1; else y=0;

   if(x==1 && y ==1){

      return 1;

   }

   else{

      return 0;

   }

   printf("%d",x);

   printf("%d\n",y);

}

int OpInv(char x){

   if(x=='V')x=0;

   if(x=='F')x=1;

   printf("%d",x);

   return x;

}

int valor(char \*e) {

   Pilha P = pilha(256);

   for(int i=0; e[i]; i++){

      if( e[i]=='V' || e[i]=='F' ) {

         printf("%c",e[i]);

         empilha(e[i],P);

      }

      else {

         if(e[i]=='!'){

            int x = desempilha(P);

            empilha(OpInv(x),P);

         }

         int y = desempilha(P);

         int x = desempilha(P);

         switch( e[i] ) {

            case '|': empilha(Opou(x,y),P); break;

            case '&': empilha(Ope(x,y),P); break;

         }

      }

   }

   int z = desempilha(P);

   destroip(&P);

   return z;

}

int prio(char o) {

   switch( o ) {

      case '|': return 0;

      case '&': return 1;

      case '!': return 2;

   }

   return -1; // operador inválido!

}

char \*posfixa(char \*e) {

   static char s[256];

   int j = 0;

   Pilha P = pilha(256);

   for(int i=0; e[i]; i++)

      if( e[i]=='(' ) empilha('(',P);

      else if( e[i]=='V' || e[i]=='F' ) s[j++] = e[i];

      else if( strchr("!&|",e[i]) ) {

         while( !vaziap(P) && prio(topo(P))>=prio(e[i]) )

            s[j++] = desempilha(P);

         empilha(e[i],P);

      }

      else if( e[i] == ')' ) {

         while( topo(P)!='(' )

           s[j++] = desempilha(P);

         desempilha(P);

      }

      while( !vaziap(P) )

         s[j++] = desempilha(P);

      s[j] = '\0';

      destroip(&P);

    return s;

}

int main(void) {

   char e[513];

   printf("Infixa booleana? ");

   gets(e);

   printf("Posfixa: %s\n", posfixa(e));

   printf("Resultado: %d", valor(posfixa(e)));

   return 0;

}



Ex: 005

#include <stdio.h>

int main(void) {

    int n;

    printf("Numero? ");

    scanf("%d",&n);

    int s=0;

    while( n>0 ) {

        s += n%10;

        n /= 10;

    }

    printf("Soma dos digitos = %d\n",s);

    return 0;

}



Ex: 006

#include <stdio.h>

#include <time.h>

#include <stdlib.h>

int main(void) {

    srand(time(NULL));

    int c, n = rand()%7 + 1;

    do {

        printf("Chute entre 1 e 7: ");

        scanf("%d",&c);

        if( c<n ) puts("Baixo!");

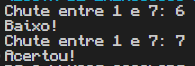
        else if( c>n ) puts("Alto!");

    } while( n!=c );

    puts("Acertou!");

    return 0;

}



Ex: 007

#include <stdio.h>

int fat(int n) {

    int f = 1;

    for(int i=2; i<=n; i++) f \*= i;

    return f;

}

int main(void) {

    printf("Fatorial do 5: %d\n", fat(5));

    return 0;

}



Ex:008

#include <stdio.h>

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

    for(int i=0; i<n; i++) {

        for(int j=0; j<v[i]; j++)

        putchar(220);

        putchar('\n');

    }

}

int main(void) {

    int a[4] = {3,4,2,1};

    int b[3] = {9,4,7};

    barras(a,4);

    getchar();

    barras(b,3);

    return 0;

}



Ex: 009

#include <stdio.h>

#include <string.h>

int main(void) {

    char s[256];

    printf("Senha? ");

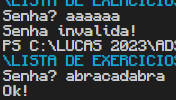
    gets(s);

    if( strcmp(s,"abracadabra")==0 ) puts("Ok!");

    else puts("Senha invalida!");

    return 0;

}



Ex: 010

#include <stdio.h>

typedef struct

{

    float x;

    float y;

} Ponto;

int main(void)

{

    Ponto p = {

        1.5,

        2.5};

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

    return 0;

}



Ex: 011

#include <stdio.h>

int main(void) {

    int v = 5;  // variável simples

    int \*p;     // variável ponteiro

    p = &v;

    \*p = \*p + 2;

    printf("v=%d, \*p=%d\n",v,\*p);

    return 0;

}



Ex: 012

#include <stdio.h>

void troca(int a, int b) {

    int c = a;

    a = b;

    b = c;

}

int main(){

    int x = 5, y=3;

    troca(x,y);

    printf("o valor de x= %d, e o valor de y= %d", x,y);

    return 0;

}



Ex: 013

#include <stdio.h>

void troca(int \*a, int \*b) {

    int c = \*a;

    \*a = \*b;

    \*b = c;

}

int main(){

    int x = 5, y=3;

    troca(&x, &y);

    printf("o valor de x= %d, e o valor de y= %d", x,y);

    return 0;

}



Ex: 014

#include <stdio.h>

#include <stdlib.h>

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

    float s = 0;

    for(int i=0; i<n; i++)

        s += v[i];

    return s/n;

}

int main(void){

    int n;

    printf("Quantidade de numeros? ");

    scanf("%d",&n);

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

    for(int i=0; i<n; i++) {

        printf("%do numero? ",i+1);

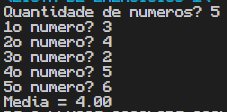
        scanf("%f",&v[i]);

    }

    printf("Media = %.2f\n",media(v,n));

    return 0;

}



Ex: 015

#include <stdio.h>

#include <stdlib.h>

typedef struct no \*Ptr;

struct no {

    int item;

    Ptr prox;

};

Ptr no(int x, Ptr p) {

    Ptr n = malloc(sizeof(struct no));

    n->item = x;

    n->prox = p;

    return n;

}

int main(void) {

    Ptr p = no(3,no(1,no(5,NULL)));

    while( p != NULL ) {

        printf("%d\n",p->item);

        p = p->prox;

    }

    return 0;

}

