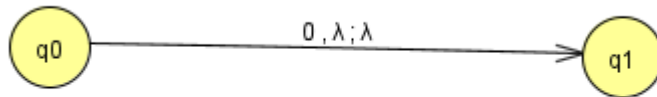


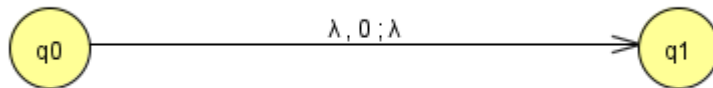
Simulador de PDA, Representação gráfica

linguagem de entrada = $\{0,1\}$.

primeiro irá pedir o PUSH.



depois o pop



Depois ira executar esse passo ate esvaziar o Automato completamente.

linguagem de entrada = $\{0,1\}$.

<https://github.com/SirSaito/APP.git>

```
/*
Descrição do Trabalho de Automato e Pilha:

"Desenvolver um AAP (Autômato a Pilha), ele deve simular o
funcionamento teórico de um autômato.

Requisitos mínimos:

Uma cadeia de entrada (definida pelo alfabeto desejado) e dizer se
reconhece ou não
Definir em código toda a sêxtupla (mesmo que não utilize todas as
variáveis)
Ideias de extras:

Fazer a definição dos estados dinâmicos
Definir a linguagem
Definir alfabeto, dentre outros elementos da sêxtupla como entrada.
BONUS: Fazer representação gráfica"
```

```

*/

#include<stdio.h>    // biblioteca stdio.
#include<stdlib.h>   // biblioteca stdlib.
#include<locale.h>   // biblioteca locale.

typedef struct NO{    // Estrutura do interior da Pilha.
    int dado;        //
    struct NO *prox;    //
} NO;

typedef struct PILHA{ //Topo da Pilha.
    NO *topo;        // Tamanho da pilha.
} PILHA;

void inicializaPilha(PILHA *p){ //Topo aponta para o primeira
referencia da Pilha.
    p -> topo = NULL;
}

void empilha(int dado, PILHA *p){ //cria espaco na Pilha e coloca o
primeiro elemento.
    NO *ptr = (NO*) malloc(sizeof(NO));

    if(ptr == NULL){
        printf("/t/tErro de alocao de novo no./n");
        return;
    }

    else{
        ptr -> dado = dado;        //ptr o espaco da pilha aponta para
dado.
        ptr -> prox = p -> topo;    //prox do elemnto aponta pata o
topo.
        p -> topo = ptr;        //elemento enserido no topo.
    }
}

int desempilha(PILHA *p){ //Aponta para dado para remover elemento.
    NO* ptr = p -> topo; //ptr apontando para o elemento no topo da
Pilha.
    int dado;

```

```

    if(ptr == NULL){ //Checagem se a Pilha esta vazia.
        printf("\t\tPilha vazia.\n");
        return 0;
    }

    else{ //Desempilha Elemento.
        p -> topo = ptr -> prox;
        ptr -> prox = NULL;
        dado = ptr -> dado;
        free (ptr);
        return dado;
    }
}

void imprimePilha(PILHA *p){ //Imprime a Pilha.
    NO *ptr = p -> topo;

    if(ptr == NULL){ //Checagem se a Pilha esta vazia.
        printf("\t\tPilha vazia.\n");
        return;
    }

    else{
        while (ptr != NULL){
            printf("\t\t%d\n", ptr -> dado);
            ptr = ptr -> prox;
        }
    }
}

int main(){
    PILHA *p1 = (PILHA*) malloc(sizeof(PILHA)); //Chama Pilha no
Main().
    int dado, i, x, y, o, r;
    int m = 0;

    printf("\t\tBem vindo ao simulador de Automato a Pilha.\n");
//Mensagem de Bem vindo.
    printf("\t\tPara fins praticos o alfabeto que sera usado sera
binario (0/1)\n"); //Mensagem de definicao de linguagem do Automato.
    printf("\t\tAtualmente estamos no ESTADO INICIAL, siginifica que a
Pilha esta vazia\n"); //Mensagem de definicao do estado inicial.

```

```

    printf("\t\tA seguir sera pedido para enserir elementos na Pilha
quando colocados se chama os ESTADOS DE TRANSICAO\n");    //Mensagem de
definicao dos ESTADOS DE TRANSICAO.

    printf("\t\tVoce podera manipular a Pilha colocando um
elemento(PUSH), tirando(POP) ou nada( $\epsilon$ )\n");    //Mensagem de definicao
de Manipulacao da pilha com PUSH, POP e  $\epsilon$ .


    if (p1 == NULL){
        printf("/t/tErro de alocao da Pilha./n");
        exit(0);
    }

    else{

        inicializaPilha(p1);

        printf("\t\tQuer usar o modo com uma cadeia especifica, ou usar
o simulador livremente?(Digite 1 para primeira opcao e 0 para segunda
opcao)\n");

        printf("\t\tAVISO!!!! Assim que a escolha for feita, para poder
experenciar a outra opcao nessecita-se reiniciar o programa.\n");
        scanf("%d", &r);


        if((r != 0) && (r != 1)){
            printf("\t\tNao aceita.\n");
            while((r != 0) && (r!= 1)){
                printf("\t\tDigite de novo\n");
                scanf("%d", &r);
            }
        }

        if(r == 1){
            while(x != 0){
                printf("\t\tCadeias = [101, 1110, 1101,
1001]\n");

                printf("\t\tDeseja alocar um elemento na
pilha(PUSH)?(Digite 1 para sim e 0 para nao)\n");
                scanf("%d", &o);
                if(o == 1){

```

```

printf("\t\tDigite um elemento para alocar
na pilha(PUSH)\n");

scanf("%d", &dado);

if(dado == 1110){
    empilha (1110, p1);
    m++;
    imprimePilha(p1);
}

if(dado == 101){
    empilha (101, p1);
    m++;
    imprimePilha(p1);
}

if(dado == 1101){
    empilha (1101, p1);
    m++;
    imprimePilha(p1);
}

if(dado == 1001){
    empilha (1001, p1);
    m++;
    imprimePilha(p1);
}

if((dado != 1110) && (dado != 101) && (dado
!= 1101) && (dado != 1001)){
    printf("\t\tCadeia rejeitada\n");
}

printf("\t\tDeseja desempilhar(POP)?(Digite 1
para sim e 0 para nao)\n");
scanf("%d", &y);
if((y == 1) && (m > 0)){
    printf("\t\tTentando desempilhar(POP) -
resultado: %d\n", desempilha(p1));
    imprimePilha(p1);
    m--;
}
else{

```

```

        if((y == 1) && (m <= 0)){
            printf("\t\tFalha ao desempilhar.\n");
        }
    }
    printf("\t\tDeseja continuar?(Digite 1 para sim
e 0 para nao)\n");

    scanf("%d", &x);
    imprimePilha(p1);
}

}

if(r == 0){
while(x != 0){
    printf("\t\tDeseja alocar um elemento na
pilha(PUSH)?(Digite 1 para sim e 0 para nao)\n");
    scanf("%d", &o);
    if(o == 1){
        printf("\t\tDigite um elemento para alocar na
pilha(PUSH)\n");

        scanf("%d", &dado);
        if((dado != 0) && (dado != 1)){
            printf("\t\tPilha nao aceita.\n");
            while((dado != 0) && (dado != 1)){
                printf("\t\tDigite um elemento para ser
alocado na pilha (PUSH)\n");

                scanf("%d", &dado);
            }
        }
        empilha (dado, p1);
        m++;
        imprimePilha(p1);
    }
    printf("\t\tDeseja desempilhar(POP)?(Digite 1 para
sim e 0 para nao)\n");
    scanf("%d", &y);
    if((y == 1) && (m > 0)){
        printf("\t\tTentando desempilhar(POP) -
resultado: %d\n", desempilha(p1));
        imprimePilha(p1);
        m--;
    }
    else{
        if((y == 1) && (m <= 0)){

```

```

        printf("\t\tFalha ao desempilhar.\n");
    }
}

printf("\t\tDeseja continuar?(Digite 1 para sim e 0
para nao)\n");

scanf("%d", &x);
imprimePilha(p1);
}

}

if(m > 0){
    printf("\t\tElementos da pilha:\n");
    imprimePilha(p1);
    for(i=0 ; i < m ; i++){
        printf("\t\tTentando desempilhar(POP) - resultado:
%d\n", desempilha(p1));
        imprimePilha(p1);
    }
}

printf("\t\tAutomato vazio, Pilha aceita, tenha um bom
dia!!!\n");
}

return 0;
}

```