

UERJ-ZO

Disciplina: Estrutura de Dados I

Professor: Raul Carlos Costa

Nome da Aluna: Karen Gomes / Matrícula: 2211312138

Nome do Aluno: Jonatha Salles Menezes / Matrícula: 2211312125

Nome do Aluno: Anthony Raphael / Matrícula:2211312123

Trabalho AV1 – Estrutura de Dados

Rio de Janeiro / RJ

Dezembro, 2022

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#include <time.h>

#define TAM_VET 10
#define VERD 1
#define FALSO 0
#define ERRO -1
#define VERMELHO "\033[31m"
#define CIANO "\033[36m"
#define VERDE "\033[32m"
#define AMARELO "\033[33m"
#define ROXO "\033[35m"
#define LIMPA "\033[m"
#define INV "\033[7m"

int main();
float sleep();

struct info{
    int inicio;
    int termino;
    int tamanho;
    int elemento[10];
};
typedef struct pilha{
    int dado[TAM_VET];
    int topo;
}Pilha;

int menu()
{
    int escolha;
```

```

do {
    system("cls");
    printf("=====");
    printf("\n%sEscolha uma opcao:%s \n", INV, LIMPA);
    printf("%s[1]Fila%s \n", CIANO, LIMPA);
    printf("%s[2]Pilha%s \n", AMARELO, LIMPA);
    printf("%s[3]Encerrar%s\n%s-->%s ", VERMELHO, LIMPA, ROXO, LIMPA);
    scanf("%d", &escolha);
} while (escolha != 1 && escolha != 2 && escolha != 3);
return escolha;
}

```

```

void inicia_fila(struct info* F)
{
    F->inicio = 0;
    F->tamanho = 0;
    F->termino = -1;
}

```

```

int fila_vazia(struct info* F)
{
    if(F->tamanho > 0) return FALSO;
    return VERD;
}

```

```

int fila_cheia(struct info* F)
{
    if (F->tamanho < 10) return FALSO;
    return VERD;
}

```

```

void inserir_fila(struct info* F)
{
    printf("Digite o elemento a ser inserido na FILA \n%s-->%s ", ROXO, LIMPA);
}

```

```

F->termino++;
scanf("%i", &F->elemento[F->termino]);
F->tamanho++;
}

void remover_fila(struct info* F)
{
    printf("%sElemento removido e:%s %d\n", VERDE, LIMPA, F->elemento[F->inicio]);
    F->inicio++;
    F->tamanho--;
}

void mostrar_fila(struct info* F)
{
    int i;
    printf("\n%sFila: [", CIANO);
    for(i = F->inicio; i <= F->termino; i++){
        printf("%i", F->elemento[i]);
        if(i != F->termino)
            printf(", ");
    }
    printf("]%s\n", LIMPA);
    printf("Inicio da fila esta na posicao: %d\n", F->inicio);
    printf("Termino da fila esta na posicao: %d\n", F->termino);
    printf("O tamanho da fila e: %d\n", F->tamanho );
    printf("\n");
    sleep(3);
}

int submenu_fila()
{
    int opcao;
    do{
        system("cls");

```

```

printf("=====\n");
printf("%sO que deseja
fazer?\n%s[1]Adicionar\n%s[2]Remover\n%s[3]Mostrar\n%s[4]Reiniciar\n[5]
Voltar\n%s-->%s ",
INV, LIMPA, VERDE, LIMPA, VERMELHO, LIMPA, CIANO, LIMPA,
AMARELO, LIMPA, ROXO, LIMPA);
scanf("%i", &opcao);
}while (opcao < 0 || opcao > 5);
return opcao;
}

```

```

void filaFunc ()
{
system("cls");
printf("%sCarregando fila...%s", VERMELHO, LIMPA);
sleep(1);
struct info F;
int opcao;
int ok;
inicia_fila(&F);
while(1){ //Loop infinito
opcao = submenu_fila();
if(opcao == 5) break; //Quebra no início para evitar o else
if(opcao == 1){
ok = fila_cheia(&F);
system("cls");
if(ok == VERD)
printf(" %sFila Cheia!!!%s ", VERMELHO, LIMPA);
else
inserir_fila(&F);
printf(" %sElemento inserido com sucesso!!!%s ", VERDE, LIMPA);
sleep(1);
}
}
}

```

```

    }

    else if(opcao == 2){
        system("cls");
        ok = fila_vazia(&F);
        if(ok == VERD)
            printf("%sRemocao invalida! Fila vazia\n%s", VERMELHO, LIMPA);
        else
            remover_fila(&F);
        sleep(1);
    }

    else if(opcao == 3){
        system("cls");
        mostrar_fila(&F);
    }

    else{
        system("cls");
        inicia_fila(&F);
        printf("%sFila reiniciada.%s\n", CIANO, LIMPA);
        sleep(1);
    }

}

main();
}

void pilhaFunc()
{
    Pilha p1;
    int op = -1;
    int dado, l;
    inicializaPilha(&p1);

```

```

while( op!= 0 ) {
    printf( "\n\nMenu\n1-Inserir\n2-Excluir\n3-Exibir Vetor\n4-Reinicia
Pilha\n0-Fim\nOpcao: ");
    scanf("%d", &op);
    if( op== 1) {
        printf( "\nDigite um número: ");
        scanf("%i", &dado);
        empilha(dado, &p1);
    }
    else if( op== 2 ) {
        desempilha(&p1);
    }
    else if( op== 3 ) {
        imprimePilha(&p1);
    }
    else if( op == 4){
        inicializaPilha(&p1);
    }
}
main();
}

void inicializaPilha(Pilha *p){
    int i;
    for(i=0; i < TAM_VET; i++){
        p->dado[i] = 0;
    }
    p->topo = 0;
}

void empilha(int dado, Pilha *p){
    if (p->topo == TAM_VET){
        printf("Pilha cheia.\n");
        return;
    }
}

```

```

    } else {
        p->dado[p->topo] = dado;
        p->topo++;
        return;
    }
}

```

```

int desempilha(Pilha *p){
    int aux;
    if (p->topo == 0){
        printf("Pilha Vazia.\n");
        return ;
    } else {
        aux = p->dado[p->topo - 1];
        printf("\nElemento removido : %i ", aux);
        p->topo--;
        return p->dado[p->topo];
    }
}

```

```

void imprimePilha(Pilha *p){
    int i;
    for(i=0; i<p->topo; i++){
        printf("%d", p->dado[i]);
        printf("\n");
    }
}

```

```

int main()
{
    system("cls");
    setlocale(LC_ALL, "");
    int escolha;
    escolha = menu();
}

```



```
    if (escolha == 1)
        filaFunc();

    else if(escolha == 2)
        pilhaFunc();

    //O programa só chega até aqui com a escolha 3
    system("cls");
    printf("\n\tPrograma encerrado.\033[30m");

    return 0;
}
```