

Centro Universitário Nobre - Unifan
Disciplina: Algoritmo e Estrutura de Dados
Docente: Cleide Bittencourt
Discentes: Gabriel Alves Moraes, Matheus Oliveira
Curso: Análise e Desenvolvimento de Sistemas (Noturno)

Atividade Fila e Pilha

Questão 01

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#define MAX 100

int pilha[MAX];
int topo;

//Iniciar a pilha
void iniciar(){
    topo = -1;
}

// Inserir elemento na pilha
void empilhar(int elemento) {
    topo++;
    pilha[topo] = elemento;
}

// Remover elemento da pilha
int desempilhar() {
    return pilha[topo--];
}

// Verificar se a pilha está vazia
int pilhaVazia() {
    if(topo == -1){
        return 1;
    }
    else{
        return 0;
    }
}

// Função para converter decimal para binário OK
void decimalParaBinario(int numeroDecimal) {
    while (numeroDecimal > 0) {
        empilhar(numeroDecimal % 2);
        numeroDecimal /= 2;
    }

    // Desempilhar e exibir o resultado binário
    printf("O resultado binário é: ");
    while (pilhaVazia() == 0) {
        printf("%d", desempilhar());
    }
}
```

```

    printf("\n");
}

int main() {
    setlocale(LC_ALL, "");
    iniciar();

    int numero;

    printf("Digite um número inteiro positivo: ");
    scanf("%d", &numero);

    if (numero < 0) {
        printf("Por favor, insira um número inteiro positivo.\n");
        return 1;
    }

    decimalParaBinario(numero);

    return 0;
}

```

Questão 02

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<locale.h>
#define MAX 10

int filaPar[MAX];
int filaImpar[MAX];
int pilha[MAX];
int comecoPar, finalPar, comecoImpar, finalImpar, topo;

//Inicia a fila PAR
void iniciarPar(){
    comecoPar = 0;
    finalPar = 0;
}

//Inicia a fila IMPAR
void iniciarImpar(){
    comecoImpar = 0;
    finalImpar = 0;
}

//Iniciar Pilha
void iniciarPilha(){
    topo = -1;
}

//Inserir elemento na FILA PAR
void inserirPar(int elemento){
    filaPar[finalPar] = elemento;
    finalPar++;
}

//Inserir elemento na FILA IMPAR
void inserirImpar(int elemento){
    filaImpar[finalImpar] = elemento;
    finalImpar++;
}

```

```

//Remover elemento da FILA PAR
int removerPar(){
    return filaPar[comecoPar++];
}

//Remover elemento da FILA IMPAR
int removerImpar(){
    return filaImpar[comecoImpar++];
}

//Inserir elemento na pilha
void empilhar(int elemento){
    if (topo == MAX - 1) {
        printf("Pilha cheia! Não é possível empilhar.\n");
        return;
    }
    pilha[++topo] = elemento;
}

//Remover elemento da pilha
int desempilhar() {
    if (topo == -1) {
        printf("Pilha vazia! Não é possível desempilhar.\n");
        return -1;
    }
    return pilha[topo--];
}

int main(){
    setlocale(LC_ALL, "");

    iniciarPar();
    iniciarImpar();
    iniciarPilha();

    int numero;

    // Leitura dos valores
    printf("Digite os valores inteiros (digite 0 para finalizar):\n");
    do {
        scanf("%d", &numero);

        if (numero != 0) {
            // Determina se o número é par ou ímpar
            if (numero % 2 == 0) {
                inserirPar(numero);
            } else {
                inserirImpar(numero);
            }
        }
    } while (numero != 0);

    // Remover elementos das filas e inserir na pilha alternadamente
    while (comecoPar < finalPar || comecoImpar < finalImpar) {
        if (comecoImpar < finalImpar) {
            int elementoImpar = removerImpar();
            printf("Retirado da FILA ÍMPAR: %d\n", elementoImpar);
            empilhar(elementoImpar);
        }

        if (comecoPar < finalPar) {
            int elementoPar = removerPar();
            printf("Retirado da FILA PAR: %d\n", elementoPar);
        }
    }
}

```

```

        if (elementoPar > 0) {
            empilhar(elementoPar);
        } else {
            int elementoRemovido = desempilhar();
            printf("Removido da PILHA: %d\n", elementoRemovido);
        }
    }
}

// Mostrar o conteúdo da pilha
printf("\nConteúdo final da PILHA:\n");
while (topo >= 0) {
    printf("%d\n", desempilhar());
}

return 0;
}

```

Questão 03

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<locale.h>
#define MAX 10

int pilha[MAX];
int topo;
int fila[MAX], filaiv[MAX];
int comeco, final, comecolv, finallv;

//Iniciar Pilha
void iniciarPilha(){
    topo = -1;
}

//Inserir elemento
void inserirPilha(int elemento){
    topo++;
    pilha[topo] = elemento;
}

//Remover elemento
void removerPilha(){
    printf("O NUMERO RETIRADO FOI: %d\n", pilha[topo]);
    topo--;
}

//Listar pilha
void listarPilha(){
    printf("A listagem dos valores da pilha é: \n");
    for(int x = 0; x<topo; x++){
        printf("%d\n", pilha[x]);
    }
}

//Iniciar fila
void iniciarFila(){
    comeco = 0;
    final = 0;
}

//Inserir elemento
void inserirFila(int elemento){

```

```

        fila[final] = elemento;
        final++;

    }

//Remover elemento
void removerFila(){
    printf("O valor retirado e: %d\n", fila[comeco]);
    comeco++;

}

// Listar fila
void listarFila() {
    printf("A listagem dos valores da fila é: \n");
    for (int x = 0; x < final; x++) {
        printf("%d\n", fila[x]);
    }
}

//Iniciar fila INVERTIDA
void iniciarFilalv(){
    comecolv = 0;
    finallv = 0;
}

//Inserir elemento FILA INVERTIDA
void inserirFilalv(int elemento){
    filalv[finallv] = elemento;
    finallv++;

}

//Remover elemento FILA INVERTIDA
void removerFilalv(){
    printf("O valor retirado e: %d\n", filalv[comecolv]);
    comecolv++;

}

// Listar FILA INVERTIDA
void listarFilalv() {
    printf("A listagem dos valores da fila INVERTIDA é: \n");
    for (int x = 0; x < finallv; x++) {
        printf("%d\n", filalv[x]);
    }
}

//Mudar de fila para pilha
void filaParaPilha(){
    for(int x = comeco; x<=final; x++){
        inserirPilha(fila[x]);
    }
}

//Mudar de Pilha para Fila Invertida
void pilhaParaFilalv(){
    for (int x = topo; x > -1; x--){
        if(pilha[x] != 0){
            inserirFilalv(pilha[x]);
        }
    }
}

```

```

int main(){
    setlocale(LC_ALL, "");
    iniciarPilha();
    iniciarFila();
    iniciarFilalv();

    int elemento;

    printf("Insira os elementos na fila [Digite 0 para parar]:\n");
    do {
        scanf("%d", &elemento);
        if (elemento != 0) {
            inserirFila(elemento);
        }
    } while (elemento != 0);
    system("cls");

    //fila para pilha
    filaParaPilha();
    //fila
    listarFila();
    //pilha
    listarPilha();
    //pilha para fila invertida
    pilhaParaFilalv();
    //listar fila invertida
    listarFilalv();

}

```

Questão 04

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<locale.h>
#define MAX 10

int pilha[MAX];
int topo;

//Inicia a pilha
void iniciar(){
    topo = -1;
}

//Ver se a pilha está cheia
int verCheia(){
    if(topo==MAX){
        printf("Pilha cheia!\n");
        return 1;
    }
    else{
        return 0;
    }
}

//Ver se a pilha está vazia
int verVazia(){
    if(topo == -1){
        printf("A pilha está vazia!\n");
        return 1;
    }
}

```

```

        else{
            return 0;
        }
    }

//Inserir elemento
void inserir(int elemento){
    topo++;
    pilha[topo] = elemento;
}

//Remover elemento
void remover(){
    printf("O NUMERO RETIRADO FOI: %d\n", pilha[topo]);
    topo--;
}

//Listar pilha
void listar(){
    printf("A listagem dos valores da pilha é: \n");
    for(int x = 0; x<=topo; x++){
        printf("O elemento na posição %d é: %d\n",x,pilha[x]);
    }
}

//Ver o topo da pilha
void verTopo(){
    printf("O topo da pilha é: %d\n",pilha[topo]);
}

int main(){
    setlocale(LC_ALL,"");
    iniciar();
    int elemento,tirar;

    //Inserindo valores na fila
    printf("Insira os elementos na fila [Digite 0 para parar]:\n");
    do {
        scanf("%d", &elemento);
        if (elemento != 0) {
            inserir(elemento);
        }
    } while (elemento != 0);
    system("cls");

    //mostrar pilha
    listar();

    //recolher a quantidade que vai remover da pilha
    printf("Digite a quantidade de numeros que deseja remover da pilha: \n");
    scanf("%d",&tirar);
    for(int x = 0; x < tirar; x++){
        remover();
    }
    printf("\n");

    //mostrar pilha com os elementos removidos
    listar();

}

```

Questão 05

```

#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
#define MAX 4

int fila[MAX];
int comeco, final, qtd;

//Inicia a fila
void iniciar(){
    comeco = 0;
    final = 0;
    qtd = 0;
}

//Ver cheia
int verCheia(){
    if(qtd == MAX){
        return(1);
    }
    return(0);
}

//Ver vazia
int verVazia(){
    if(qtd == 0){
        return(1);
    }
    return(0);
}

//Inserir valor
void inserir(int valor){
    if(verCheia() == 1){
        verCheia();
    }
    else{
        if((final == MAX) && (qtd != MAX)){
            final = 0;
            fila[final] = valor;
            final++;
            qtd++;
        }
        else{
            fila[final] = valor;
            final++;
            qtd++;
        }
    }
}

//Remover valor
void remover(){
    if(verVazia() == 1){
        verVazia();
    }
    else{
        if((final < comeco) && (qtd != 0)){
            comeco = 0;
            printf("O valor retirado é: %d\n", fila[comeco]);
            comeco++;
            qtd--;
        }
    }
}

```



```

        else{
            printf("O valor retirado é: %d\n",fila[comeco]);
            comeco++;
            qtd--;
        }
    }
}

//Listar
void listar(){
    if(comeco < final){
        printf("Os valores da fila é:\n");
        for(int y = comeco; y < final; y++){
            printf("%d\n",fila[y]);
        }
    }
    else{
        printf("Os valores da fila é:\n");
        for(int x = comeco; x <= MAX -1; x++){
            printf("%d\n",fila[x]);
        }
        for(int x = 0; x <= final -1; x++){
            printf("%d\n",fila[x]);
        }
    }
}

//Ver Inicio
void inicio(){
    if(verVazia() == 1){
        verVazia();
    }
    else{
        printf("O TOPO da fila é:%d\n",fila[comeco]);
    }
}

//Ver fim
void fim(){
    if(verVazia() == 1){
        verVazia();
    }
    else{
        printf("O FIM da fila é: %d\n",fila[final -1]);
    }
}

void menu() {
    int escolha;

    int valor;

    do{
        system ("cls");

        printf("Menu:\n");
        printf("1. Inserir valor \n");
        printf("2. Remover valor \n");
        printf("3. Ver cheia \n");
        printf("4. Ver vazia\n");
        printf("5. Ver inicio\n");
        printf("6. Ver fim\n");
        printf("7. Listar\n");
        printf("8. Sair\n");
    }
}

```

```

printf("Escolha a opção\n");
scanf("%d", &escolha);

if(escolha >= 1 && escolha <= 9){
    switch (escolha) {
case 1:
    printf("Digite o valor:\n");
    scanf("%d",&valor);
    inserir(valor);
    break;
case 2:
    remover();
    break;
case 3:
    if(verCheia()==1){
        verCheia();
        printf("A fila está cheia !\n");
    }
    else{
        verVazia();
        printf("A fila não está cheia !\n");
    }

    break;
case 4:
    if(verVazia()==1){
        verVazia();
        printf("A fila está vazia !\n");
    }
    else{
        verCheia();
        printf("A fila não está vazia!\n");
    }

    break;
case 5:
    inicio();
    break;
case 6:
    fim();
    break;

case 7:
    listar();
    break;

    }
    } else {
        printf("Opção inválida. Tente novamente?");
    }

    printf("Pressione enter para continuar...");
    getchar();
    getchar();

}while(escolha != 8);

}

int main(){
    setlocale(LC_ALL, "portuguese");
    menu();
    return 0;

}

```

Questão 06

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<locale.h>
#define MAX 10

int fila[MAX];
int filaPar[MAX];
int filaImpar[MAX];
int comeco, final, comecoPar, finalPar, comecoImpar, finalImpar;

//Inicia a FILA
void iniciar(){
    comeco = 0;
    final = 0;
}

//Inicia a fila PAR
void iniciarPar(){
    comecoPar = 0;
    finalPar = 0;
}

//Inicia a fila IMPAR
void iniciarImpar(){
    comecoImpar = 0;
    finalImpar = 0;
}

//Inserir elemento na FILA
void inserir(int elemento){
    fila[final] = elemento;
    final++;
}

//Inserir elemento na FILA PAR
void inserirPar(int elemento){
    filaPar[finalPar] = elemento;
    finalPar++;
}

//Inserir elemento na FILA IMPAR
void inserirImpar(int elemento){
    filaImpar[finalImpar] = elemento;
    finalImpar++;
}

//Remover elemento na FILA
void remover(){
    comeco++;
}

//Remover elemento da FILA PAR
void removerPar(){
    comecoPar++;
}

//Remover elemento da FILA IMPAR
void removerImpar(){
    comecoImpar++;
}
```

```

//Mostrar FILA
void listar() {
    printf("A listagem dos valores da fila é: \n");
    for (int x = 0; x < final; x++) {
        printf("%d\n", fila[x]);
    }
}

//Mostrar fila PAR
void listarPar() {
    printf("A listagem dos valores da fila PAR é: \n");
    for (int x = 0; x < finalPar; x++) {
        printf("%d\n", filaPar[x]);
    }
}

//Mostrar fila IMPAR
void listarImpar() {
    printf("A listagem dos valores da fila IMPAR é: \n");
    for (int x = 0; x < finalImpar; x++) {
        printf("%d\n", filaImpar[x]);
    }
}

int main(){
    setlocale(LC_ALL, "");

    iniciarPar();
    iniciarImpar();
    iniciar();

    int numero;

    // Leitura dos valores
    printf("Digite os valores inteiros (digite 0 para finalizar):\n");
    do {
        scanf("%d", &numero);
        inserir(numero);

    } while (numero != 0);
    system("cls");
    listar();//mostrar FILA

    for(int x = comeco; x < final; x++){//Dividir a fila principal em fila par e impar
        if (fila[x] != 0){
            // Determina se o número é par ou ímpar
            if (fila[x] % 2 == 0){
                inserirPar(fila[x]);
            }
            else{
                inserirImpar(fila[x]);
            }
        }
    }

    listarPar();
    printf("\n");
    listarImpar();

    return 0;
}

```

Questão 07

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 50

//Estrutura da pilha
typedef struct {
    char itens[MAX];
    int topo;
} Pilha;

//Inicia a pilha
void inicializarPilha(Pilha *pilha) {
    pilha->topo = -1;
}

//Ver se a pilha esta vazia
int pilhaVazia(Pilha *pilha) {
    return (pilha->topo == -1);
}

//Ver se a pilha esta cheia
int pilhaCheia(Pilha *pilha) {
    return (pilha->topo == MAX - 1);
}

//Inserir valor na pilha
void empilhar(Pilha *pilha, char caractere) {
    if (pilhaCheia(pilha)){
        printf("A pilha está cheia! Não é possível empilhar.\n");
        return;
    }
    pilha->itens[++(pilha->topo)] = caractere;
}

//Remover valor da pilha
char desempilhar(Pilha *pilha) {
    if (pilhaVazia(pilha)) {
        printf("A pilha está vazia! Não é possível desempilhar.\n");
        return '\0';
    }
    return pilha->itens[(pilha->topo)--];
}

//Inverter a string usando uma pilha
void inverterString(char *entrada) {
    Pilha pilha;
    inicializarPilha(&pilha);

    int tamanho = strlen(entrada); //Retorna o numero inteiro de uma string
    for (int i = 0; i < tamanho; i++) {
        empilhar(&pilha, entrada[i]);
    }

    for (int i = 0; i < tamanho; i++) {
        entrada[i] = desempilhar(&pilha);
    }
}

int main() {
    char entrada[MAX];
```

```

printf("Digite a string a ser invertida: ");
fgets(entrada, sizeof(entrada), stdin);
entrada[strcspn(entrada, "\n")] = '\0'; //Remove o caractere de nova linha adicionado pelo fgets

inverterString(entrada);

printf("String invertida: %s\n", entrada);

return 0;
}

```

QUESTÃO 08

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>

#define MAX_SIZE 100 // Tamanho máximo da fila

// Estrutura para representar um paciente
typedef struct {
    char nome[50];
    int idade;
    // Outras informações do paciente podem ser adicionadas aqui
} Paciente;

// Estrutura da fila
typedef struct {
    Paciente elementos[MAX_SIZE];
    int inicio, fim;
} Fila;

// Função para inicializar a fila
void iniciarFila(Fila *fila) {
    fila->inicio = fila->fim = -1;
}

// Função para verificar se a fila está vazia
int filaVazia(Fila *fila) {
    return (fila->inicio == -1);
}

// Função para verificar se a fila está cheia
int filaCheia(Fila *fila) {
    return ((fila->fim + 1) % MAX_SIZE == fila->inicio);
}

// Função para inserir um paciente na fila
void enfileirar(Fila *fila, Paciente paciente) {
    if (filaCheia(fila)) {
        printf("A fila está cheia. Não é possível adicionar mais pacientes.\n");
        return;
    }

    if (fila->inicio == -1) {
        fila->inicio = fila->fim = 0;
    } else {
        fila->fim = (fila->fim + 1) % MAX_SIZE;
    }
}

```

```

    fila->elementos[fila->fim] = paciente;
}

// Função para remover um paciente da fila
void desenfileirar(Fila *fila) {
    if (filaVazia(fila)) {
        printf("A fila está vazia. Não há pacientes para remover.\n");
        return;
    }

    if (fila->inicio == fila->fim) {
        fila->inicio = fila->fim = -1;
    } else {
        fila->inicio = (fila->inicio + 1) % MAX_SIZE;
    }
}

// Função para obter o próximo paciente da fila (sem remover)
void proximoPaciente(Fila *fila) {
    if (filaVazia(fila)) {
        printf("A fila está vazia. Não há pacientes no momento.\n");
        return;
    }

    printf("Próximo paciente a ser atendido: %s, %d anos.\n",
        fila->elementos[fila->inicio].nome,
        fila->elementos[fila->inicio].idade);
}

int main() {
    setlocale(LC_ALL, "portuguese");
    Fila filaDeEspera;
    iniciarFila(&filaDeEspera);

    int opcao;
    Paciente novoPaciente;

    printf("Bem-vindo à Clínica Médica São Bento\n");

    do {
        printf("\nEscolha uma opção:\n");
        printf("1. Adicionar novo paciente na fila\n");
        printf("2. Remover próximo paciente atendido\n");
        printf("3. Ver próximo paciente a ser atendido\n");
        printf("0. Sair\n");
        printf("Opção: ");
        scanf("%d", &opcao);

        switch (opcao) {
            case 1:
                printf("\nInforme o nome do paciente: ");
                scanf("%s", novoPaciente.nome);
                printf("Informe a idade do paciente: ");
                scanf("%d", &novoPaciente.idade);
                enfileirar(&filaDeEspera, novoPaciente);
                printf("Paciente adicionado na fila com sucesso.\n");
                break;
            case 2:
                desenfileirar(&filaDeEspera);
                printf("Paciente removido da fila.\n");
                break;
            case 3:
                proximoPaciente(&filaDeEspera);
                break;
        }
    } while (opcao != 0);
}

```

```
        case 0:
            printf("Encerrando o programa. Obrigado!\n");
            break;
        default:
            printf("Opção inválida. Tente novamente.\n");
    }
} while (opcao != 0);

return 0;
}
```