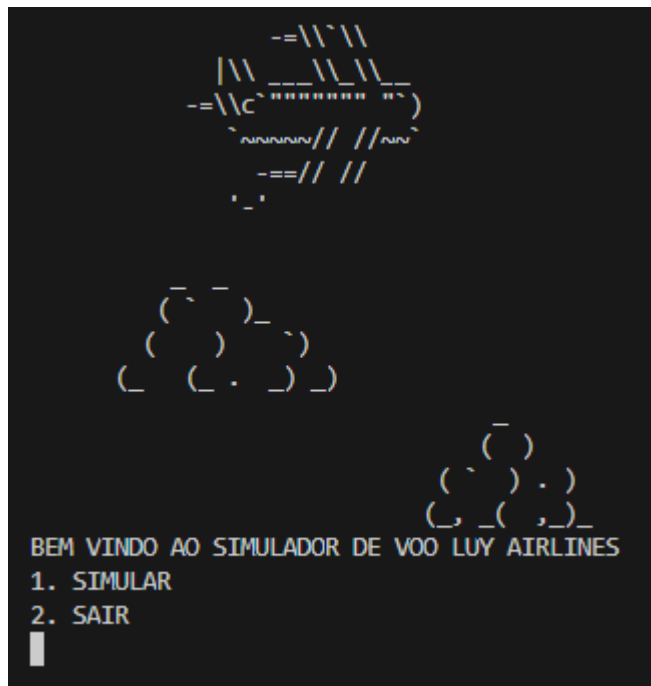


Trabalho 2 da disciplina de Algoritmos e Estrutura de Dados
Aluno: João Samuel Luy

O Algoritmo simula 50 vôos chegando e saindo de um aeroporto, são 3 pistas que levam duas unidades de tempo para atenderem completamente os aviões, tanto para decolagem como para pouso, a cada unidade de tempo chegam 4 solicitações para uso de uma das pistas, cada solicitação tem uma prioridade, indo da prioridade 1, que é urgente, para pousos por falha no motor ou falta de combustível, 2 para pousos normais e 3 para decolagens.

1. Menu

Ocorre uma pequena animação para introdução do programa juntamente com o menu



Caso digite 2, o programa irá se encerrar

Caso digite 1, o programa irá rodar a simulação

2. Simulação

Assim que inicia a simulação começam a aparecer as unidades de tempo, que apresentam a fila de vôos a serem atendidos, cada vôo com detalhes de seu número, prioridade e descrição da prioridade, em seguida, são ocupadas as pistas para atender os vôos e por último impresso a fila de vôos restantes a serem atendidos

```
----- UNIDADE DE TEMPO: 1 -----
FILA DE VOOS A SEREM ATENDIDOS:
  Numero do Voo: 1 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 2 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 3 Prioridade do Voo: 2 (Pouso normal)
  Numero do Voo: 4 Prioridade do Voo: 2 (Pouso normal)
PISTA 1 atendendo o voo 3 (Pouso normal)
PISTA 2 atendendo o voo 4 (Pouso normal)
PISTA 3 atendendo o voo 1 (Decolagem)
FILA DE VOOS A SEREM ATENDIDOS:
  Numero do Voo: 2 Prioridade do Voo: 3 (Decolagem)

----- UNIDADE DE TEMPO: 2 -----
FILA DE VOOS A SEREM ATENDIDOS:
  Numero do Voo: 2 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 5 Prioridade do Voo: 2 (Pouso normal)
  Numero do Voo: 6 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 7 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 8 Prioridade do Voo: 1 (Pouso forçado)
PISTAS EM ATENDIMENTO
```

```
----- UNIDADE DE TEMPO: 3 -----
FILA DE VOOS A SEREM ATENDIDOS:
  Numero do Voo: 2 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 5 Prioridade do Voo: 2 (Pouso normal)
  Numero do Voo: 6 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 7 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 8 Prioridade do Voo: 1 (Pouso forçado)
  Numero do Voo: 9 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 10 Prioridade do Voo: 1 (Pouso forçado)
  Numero do Voo: 11 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 12 Prioridade do Voo: 2 (Pouso normal)
PISTA 1 atendendo o voo 8 (Pouso forçado)
PISTA 2 atendendo o voo 10 (Pouso forçado)
PISTA 3 atendendo o voo 5 (Pouso normal)
FILA DE VOOS A SEREM ATENDIDOS:
  Numero do Voo: 2 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 6 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 7 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 9 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 11 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 12 Prioridade do Voo: 2 (Pouso normal)

----- UNIDADE DE TEMPO: 4 -----
FILA DE VOOS A SEREM ATENDIDOS:
  Numero do Voo: 2 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 6 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 7 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 9 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 11 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 12 Prioridade do Voo: 2 (Pouso normal)
  Numero do Voo: 13 Prioridade do Voo: 2 (Pouso normal)
  Numero do Voo: 14 Prioridade do Voo: 3 (Decolagem)
  Numero do Voo: 15 Prioridade do Voo: 2 (Pouso normal)
  Numero do Voo: 16 Prioridade do Voo: 1 (Pouso forçado)
PISTAS EM ATENDIMENTO
```

Assim, seguindo sucessivamente, atendendo 3 vôos a cada 2 unidades de tempo, até todos os vôos serem atendidos

```
----- UNIDADE DE TEMPO: 33 -----  
FILA DE VOOS A SEREM ATENDIDOS:  
  Numero do Voo: 41 Prioridade do Voo: 3 (Decolagem)  
  Numero do Voo: 48 Prioridade do Voo: 3 (Decolagem)  
PISTA 1 atendendo o voo 41 (Decolagem)  
PISTA 2 atendendo o voo 48 (Decolagem)  
TODOS OS VOOS FORAM ATENDIDOS - FILA VAZIA  
  
A SIMULACAO ACABOU
```

Ao final exibe uma mensagem dizendo que todos os vôos foram atendidos, limpando a fila, e informando que a simulação acabou.

```
#include <iostream>  
#include <ctime>  
#include <windows.h>  
  
using namespace std;  
  
struct no {  
    int info;  
    int prior;  
    no *link;  
};  
  
no *inicializaFP(no *L) {  
    L = NULL;  
    return L;  
}  
  
no *insereFP(no *L, int valor, int prior)  
{  
    no *N, *P, *ANT;  
  
    N = new no;  
    N->info = valor;  
    N->prior = prior;  
  
    if (L == NULL) {  
        L = N;  
        N->link = NULL;  
    }  
    else {
```

```

        P = L;

        while ((P != NULL) && (valor >= P->info)) {
            ANT = P;
            P = P->link;
        }
        if (P == L) {
            N->link = L;
            L = N;
        }
        else {
            ANT->link = N;
            N->link = P;
        }
    }
    return L;
}

no *removeFP(no *L, int *n, int *prior) {
    no *AUX;

    if (L != NULL) {
        *n = L->info;
        *prior = L->prior;
        AUX = L;
        L = L->link;
        delete AUX;
    }
    return L;
}

no *removePD(no *L, int *n, int *prior) {
    no *AUX, *ANT;

    for (int prioridade = 1; prioridade <= 3; prioridade++) {
        AUX = L;
        ANT = NULL;

        while (AUX != NULL && AUX->prior != prioridade) {
            ANT = AUX;
            AUX = AUX->link;
        }
    }
}

```

```

        if (AUX != NULL) {
            *n = AUX->info;
            *prior = AUX->prior;

            if (ANT == NULL) {
                L = AUX->link;
            } else {
                ANT->link = AUX->link;
            }

            delete AUX;
            return L;
        }
    }

    *n = -1;
    *prior = -1;
    return L;
}

int verificaSeVazia(no *L) {
    if (L == NULL)
        return 1;
    else
        return 0;
}

void exhibe(no *L){
    no *P = L;
    while (P != NULL) {
        cout << endl << " Numero do Voo: " << P->info << " Prioridade
do Voo: " << P->prior;
        if(P->prior == 1){
            cout << " (Pouso forçado)";
        }else if(P->prior == 2){
            cout << " (Pouso normal)";
        }else{
            cout << " (Decolagem)";
        }
        P = P->link;
    }
}

```

```

void aviao() {

    Sleep(1000);
    system("cls");

    cout << R"(
        -=\\`\\
        |\\  ____\\\\_\\_
    -=\\c`""""""""" "`)
        `~~~~// //~~`
        -==// //
        ' _ '

        _ _
        ( ` ) _
        ( ) `
        ( _ ( _ . _ ) _

        _ .
        ( _ ) _
        ( _ _ ( _ , ) "
    << endl;

    Sleep(1000);
    system("cls");

    cout << R"(
        -=\\`\\
        |\\  ____\\\\_\\_
    -=\\c`""""""""" "`)
        `~~~~// //~~`
        -==// //
        ' _ '

        _ _
        ( ` ) _
        ( ) `
        ( _ ( _ . _ ) _

        _
        ( )
        ( ` ) . )
        ( _ , _ ( , _ ) _ "
    << endl;

```

```

Sleep(1000);
system("cls");

cout << R" (
    -=\\`\\
    |\\  ____\\_\\_\\_
    -=\\c`" " " " " " " " " `)
    `~~~~~// //~~~`
    -=>// //
    ' _ '

    _ _ _
    ( ` ) _
    ( ) ` )
    ( _ ( _ . _ ) _ )

    _
    ( )
    ( ` ) . )
    ( _ , _ ( , _ ) _ )

) "<< endl;

Sleep(1000);
system("cls");

cout << R" (
    -=\\`\\
    |\\  ____\\_\\_\\_
    -=\\c`" " " " " " " " " `)
    `~~~~~// //~~~`
    -=>// //
    ' _ '

    _ _ _
    ( ` ) _
    ( ) ` )
    ( _ ( _ . _ ) _ )

    _
    ( )
    ( ` ) . )
    ( _ , _ ( , _ ) _ )"

<< endl;
}

```

```

int menu(){
    int tecla;

    for(int i = 0; i<3 ; i++){
        aviao();
    }

    cout << "BEM VINDO AO SIMULADOR DE VOO LUY AIRLINES" << endl;
    cout << "1. SIMULAR" << endl;
    cout << "2. SAIR" << endl;
    cin >> tecla;
    return tecla;
}

int main(){
    no *Voos, *PD;
    int prior[51], num[51], continua = 1, ut;
    int aux1, aux2, tecla;
    Voos = inicializaFP(Voos);
    PD = inicializaFP(PD);

    unsigned seed = time(NULL);
    srand(seed);

    tecla = menu();

    if(tecla == 2){
        return 0;
    }else {

        for(int i = 1; i <= 50; i++){
            num[i] = i;
            prior[i] = rand() % 3 + 1;
            Voos = insereFP(Voos, num[i], prior[i]);
        }

        for(ut = 1; continua; ut++) {
            cout << endl;
            cout << endl << "----- UNIDADE DE TEMPO: " << ut << "
-----" << endl;

            int voosTransferidos = 0;

```



```

        while (voosTransferidos < 4) {
            if (!verificaSeVazia(Voos)) {
                Voos = removeFP(Voos, &aux1, &aux2);
                PD = insereFP(PD, aux1, aux2);
                voosTransferidos++;
            } else {
                break;
            }
        }

        cout << "FILA DE VOOS A SEREM ATENDIDOS: ";
        exhibe(PD);

        if ((ut % 2) && (continua)){
            for (int i = 1; i <= 3; i++){
                if (PD != NULL){
                    PD = removePD(PD, &aux1, &aux2);
                    if(aux2 == 1){
                        cout << endl << "PISTA " << i << "
atendendo o voo " << aux1 << " (Pouso forçado)";
                    }else if(aux2 == 2){
                        cout << endl << "PISTA " << i << "
atendendo o voo " << aux1 << " (Pouso normal)";
                    }else if(aux2 == 3){
                        cout << endl << "PISTA " << i << "
atendendo o voo " << aux1 << " (Decolagem)";
                    }
                } else {
                    cout << endl << "TODOS OS VOOS FORAM ATENDIDOS
- FILA VAZIA";

                    continua = 0;
                }
            }
            if (!verificaSeVazia(PD)){
                cout << endl << "FILA DE VOOS A SEREM ATENDIDOS: ";
                exhibe(PD);
            }
        } else {
            cout << endl << "PISTAS EM ATENDIMENTO";
        }
    }

    cout << endl << endl << "A SIMULACAO ACABOU" << endl << endl;

```

```
        tecla = 2;
    }

    return 0;
}
```