

Enzo Galeazzo Casagrande - 32209606

Marcos Carvalho Junior - 32234120

Rafael Marques Cruz Russo - 32220294

Problema das Escadas Rolantes Duplas

Código:

Declaração de constantes, estruturas e funções do ambiente;

```
#define MAX_LEN 500

struct user
{
    int momento;
    int direcao;
};

typedef struct user usuario;

void PreparaData(char data[MAX_LEN][MAX_LEN], int line, usuario *usuario);
void fillGroup(usuario *grupo, char data[MAX_LEN][MAX_LEN]);
int escada(usuario *grupo, int n);
```

Função main():

Aqui é feita a leitura do arquivo de entrada e chamada das funções auxiliares

```
1  int main()
2  {
3      int line = 0;
4      usuario *grupo;
5
6      char data[MAX_LEN][MAX_LEN];
7      FILE *file = fopen("../escada.txt", "r");
8      if (file == NULL)
9      {
10         printf("ERRO AO ABRIR O ARQUIVO");
11         return 1;
12     }
13
14     while (!feof(file) && !ferror(file))
15     {
16         if (fgets(data[line], MAX_LEN, file) != NULL)
17         {
18             line++;
19         }
20     }
21     fclose(file);
22     grupo = (usuario *)malloc(atoi(data[0]) * sizeof(usuario));
23     fillGroup(grupo, data);
24     int momento = escada(grupo, atoi(data[0]));
25     printf("Ultima chegada: %d", momento);
26 }
```

Funções fillGroup() e PrepareData():

funções auxiliares que mapeiam o arquivo de entrada em uma lista do tipo “usuario”;

```
1 void fillGroup(usuario *grupo, char data[MAX_LEN][MAX_LEN])
2 {
3     int currentLine = 0;
4     int n_pessoas = atoi(data[currentLine]);
5     currentLine++;
6     for (int i = 0; i < n_pessoas; i++)
7     {
8         PrepareData(data, i + 1, &grupo[i]);
9     }
10 }
11
12 void PrepareData(char data[MAX_LEN][MAX_LEN], int line, usuario *usuario)
13 {
14     char *token = strtok(data[line], " ");
15     if (atoi(token) == 0)
16     {
17         return;
18     }
19     usuario->momento = atoi(token);
20     token = strtok(NULL, " ");
21     usuario->direcao = atoi(token);
22 }
23
```

Função escada():

Função responsável por tomar as decisões e simular o problema

```
1  int escada(usuario *grupo, int n)
2  {
3      usuario current = grupo[0];
4
5      int i = 0, estimativa;
6
7      usuario fila[MAX_LEN];
8      int restantes = n, direcao = -1, tempo = 0;
9
10     bool await = false;
11
12     while (restantes > 0)
13     {
14         if (await && (grupo[i].momento > estimativa || i >= restantes))
15         {
16             await = false;
17             current = fila[0];
18             tempo += 10;
19             direcao = current.direcao;
20             estimativa = tempo + 10;
21             restantes -= 1;
22         }
23         else
24         {
25             current = grupo[i];
26             if (direcao == -1)
27             {
28                 tempo = current.momento < tempo ? tempo : current.momento;
29                 direcao = current.direcao;
30                 estimativa = current.momento + 10;
31
32                 i++, restantes--;
33             }
34             else if (direcao == current.direcao)
35             {
36                 tempo = current.momento;
37                 estimativa = current.momento + 10;
38                 i++, restantes--;
39             }
40             else
41             {
42                 if (grupo[i + 1].momento - grupo[i].momento > grupo[i - 1].momento)
43                 {
44                     tempo = estimativa, direcao = -1;
45                 }
46                 else if (grupo[i + 1].momento <= estimativa)
47                 {
48                     fila[0] = grupo[i];
49                     await = true;
50                     i++;
51                 }
52                 else if (grupo[i + 1].momento == 0 || grupo[i + 1].momento > estimativa)
53                 {
54                     direcao = -1;
55                 }
56             }
57         }
58     }
59     tempo += 10;
60     return tempo;
61 }
```

Teste:

Entrada:

```
escada.txt
1 3
2 5 0
3 8 0
4 13 0
```

Saída:

```
15 D:\documentos\hacken21c\30\111
Ultima chegada: 23
16 D:\documentos\hacken21c\30\111
```