

1. Descrição da solução

Nome: Luiza de Alencar

Disciplina: TPA 2021/2

Data: 06/10/21

Problema: Usar Matching, Merging (união) e Merging com repetição em 2 vetores e com 3 vetores.

Linguagem usada: Linguagem de C.

Arquivo: Os arquivos utilizados foram .c e .txt, e a forma usada foi ler os arquivos .txt e sair um arquivo de saída com o resultado.

2. Código fonte comentado

Merging em 2 vetores

```
//merging em 2 vetores
#include <stdio.h>
#include<stdbool.h>
#define MEMORIA 100000000
bool NUMERO=true;

int input(FILE *arq, int proximo){
    int flag, chave;
    flag=fscanf(arq, "%d", &chave); //lê o elemento no arquivo
    if(flag==EOF){
        chave=MEMORIA;
        if(proximo==MEMORIA){
            NUMERO=false; //verifica se a lista foi finalizada
        }
    }
    return chave;
}

int main (void){
    int chave1=0, chave2=0;
    FILE *arq1, *arq2, *arqsaida;
    arq1=fopen("arq1.txt", "r"); //abre o arquivo 1
    arq2=fopen("arq2.txt", "r"); //abre o arquivo 2
    arqsaida=fopen("arqsaida.txt", "w"); //cria o arquivo de saída
    chave1=input(arq1, chave2); //lê o primeiro do arquivo 1
    chave2=input(arq2, chave1); //lê o primeiro do arquivo 2
    while (NUMERO){ //enquanto existe numero no arquivo
        if(chave1<chave2){
            fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
        }
    }
}
```

```

        chave1=input(arq1, chave2); //lê o proximo do arquivo 1
    }else if(chave2<chave1){
        fprintf(arqsaida, "%d\n", chave2); //escreve no arquivo saída
        chave2=input(arq2, chave1); //lê o proximo do arquivo 2
    }else{
        fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
        chave1=input(arq1, chave2); //lê o proximo do arquivo 1
        chave2=input(arq2, chave1); //lê o proximo do arquivo 2
    }
}
}

```

Merging em 3 vetores

```

//merging com 3 vetores
#include <stdio.h>
#include <stdbool.h>
#define MEMORIA 100000000
bool NUMERO=true;

int input(FILE *arq, int proximo){
    int flag, chave;
    flag=fscanf(arq, "%d", &chave); //lê o elemento no arquivo
    if(flag==EOF){
        chave=MEMORIA;
        if(proximo==MEMORIA){
            NUMERO=false; //verifica se a lista foi finalizada
        }
    }
    return chave;
}

int main (void){
    int chave1=0, chave2=0, chave3=0;
    FILE *arq1, *arq2, *arq3, *arqsaida;
    arq1=fopen("arq1.txt", "r"); //abre o arquivo 1
    arq2=fopen("arq2.txt", "r"); //abre o arquivo 2
    arq3=fopen("arq3.txt", "r"); //abre o arquivo 3
    arqsaida=fopen("arqsaida.txt", "w"); //cria o arquivo de saída
    chave1=input(arq1, chave3); //lê o primeiro do arquivo 1
    chave2=input(arq2, chave2); //lê o primeiro do arquivo 2
    chave3=input(arq3, chave1); //lê o primeiro do arquivo 3
    while (NUMERO){ //enquanto existe numero no arquivo
        if(chave1<chave2){
            fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
            chave1=input(arq1, chave3); //lê o proximo do arquivo 1
        }else if(chave2<chave1){

```

```

        fprintf(arqsaida, "%d\n", chave2); //escreve no arquivo saída
        chave2=input(arq2, chave2); //lê o proximo do arquivo 2
    }else if(chave3<chave1){
        fprintf(arqsaida, "%d\n", chave3); //escreve no arquivo saída
        chave3=input(arq3, chave1); //lê o proximo do arquivo 2
    }else{
        fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
        chave1=input(arq1, chave3); //lê o proximo do arquivo 1
        chave2=input(arq2, chave2); //lê o proximo do arquivo 2
        chave3=input(arq3, chave1); //lê o proximo do arquivo 3
    }
}
}

```

Matching em 2 vetores

```

//matching em 2 vetores
#include <stdio.h>
#include<stdbool.h>
bool NUMERO = true;

int input (FILE *pt_arq){
    int flag, chave;
    flag=fscanf(pt_arq, "%d", &chave); //lê o elemento no arquivo
    if (flag==EOF){
        NUMERO=false; //verifica se a lista foi finalizada
    }
    return chave;
}

int main (void){
    int chave1=0, chave2=0;
    FILE *arq1, *arq2, *arqsaida;
    arq1=fopen("arq1.txt", "r"); //abre o arquivo 1
    arq2=fopen("arq2.txt", "r"); //abre o arquivo 2
    arqsaida=fopen("arqsaida.txt", "w"); //cria o arquivo de saída
    chave1=input(arq1); //lê o primeiro do arquivo 1
    chave2=input(arq2); //lê o primeiro do arquivo 2
    while (NUMERO){ //enquanto existe numero no arquivo
        if(chave1<chave2){
            chave1=input(arq1); //lê o proximo do arquivo 1
        }else if(chave2<chave1){
            chave2=input(arq2); //lê o proximo do arquivo 2
        }else{
            fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
            chave1=input(arq1); //lê o proximo do arquivo 1
            chave2=input(arq2); //lê o proximo do arquivo 2
        }
    }
}

```

```

    }
}

```

Matching em 3 vetores

```

//matching em 3 vetores
#include <stdio.h>
#include<stdbool.h>
bool NUMERO = true;

int input (FILE *pt_arq){
    int flag, chave;
    flag=fscanf(pt_arq, "%d", &chave); //lê o elemento no arquivo
    if (flag==EOF){
        NUMERO=false; //verifica se a lista foi finalizada
    }
    return chave;
}

int main (void){
    int chave1=0, chave2=0, chave3=0;
    FILE *arq1, *arq2, *arq3, *arqsaida;
    arq1=fopen("arq1.txt", "r"); //abre o arquivo 1
    arq2=fopen("arq2.txt", "r"); //abre o arquivo 2
    arq3=fopen("arq3.txt", "r"); //abre o arquivo 3
    arqsaida=fopen("arqsaida.txt", "w"); //cria o arquivo de saída
    chave1=input(arq1); //lê o primeiro do arquivo 1
    chave2=input(arq2); //lê o primeiro do arquivo 2
    chave3=input(arq3); //lê o primeiro do arquivo 3
    while (NUMERO){ //enquanto existe numero no arquivo
        if(chave1<chave2 || chave1<chave3){
            chave1=input(arq1); //lê o proximo do arquivo 1
        }else if(chave2<chave1 || chave2<chave3){
            chave2=input(arq2); //lê o proximo do arquivo 2
        }else if(chave3<chave1 || chave3<chave2){
            chave3=input(arq3); //lê o proximo do arquivo 3
        }else{
            fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
            chave1=input(arq1); //lê o proximo do arquivo 1
            chave2=input(arq2); //lê o proximo do arquivo 2
            chave3=input(arq3); //lê o proximo do arquivo 3
        }
    }
}

```

Merging chave dupla em 2 vetores

```
//merging chave dupla em 2 vetores
#include <stdio.h>
#include<stdbool.h>
#define MEMORIA 100000000
bool NUMERO=true;

int input(FILE *arq, int proximo){
    int flag, chave;
    flag=fscanf(arq, "%d", &chave); //lê o elemento no arquivo
    if(flag==EOF){
        chave=MEMORIA;
        if(proximo==MEMORIA){
            NUMERO=false; //verifica se a lista foi finalizada
        }
    }
    return chave;
}

int main (void){
    int chave1=0, chave2=0;
    FILE *arq1, *arq2, *arqsaida;
    arq1=fopen("arq1.txt", "r"); //abre o arquivo 1
    arq2=fopen("arq2.txt", "r"); //abre o arquivo 2
    arqsaida=fopen("arqsaida.txt", "w"); //cria o arquivo de saída
    chave1=input(arq1, chave2); //lê o primeiro do arquivo 1
    chave2=input(arq2, chave1); //lê o primeiro do arquivo 2
    while (NUMERO){ //enquanto existe numero no arquivo
        if(chave1<chave2){
            fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
            chave1=input(arq1, chave2); //lê o proximo do arquivo 1
        }else{
            fprintf(arqsaida, "%d\n", chave2); //escreve no arquivo saída
            chave2=input(arq2, chave1); //lê o proximo do arquivo 2
        }
    }
}
```

Merging chave duplicada em 3 vetores

```
//merging chave duplicada em 3 vetores
#include <stdio.h>
#include<stdbool.h>
#define MEMORIA 100000000
bool NUMERO=true;
```

```

int input(FILE *arq, int proximo){
    int flag, chave;
    flag=fscanf(arq, "%d", &chave); //lê o elemento no arquivo
    if(flag==EOF){
        chave=MEMORIA;
        if(proximo==MEMORIA){
            NUMERO=false; //verifica se a lista foi finalizada
        }
    }
    return chave;
}

int main (void){
    int chave1=0, chave2=0, chave3=0;
    FILE *arq1, *arq2, *arq3, *arqsaida;
    arq1=fopen("arq1.txt", "r"); //abre o arquivo 1
    arq2=fopen("arq2.txt", "r"); //abre o arquivo 2
    arq3=fopen("arq3.txt", "r"); //abre o arquivo 3
    arqsaida=fopen("arqsaida.txt", "w"); //cria o arquivo de saída
    chave1=input(arq1, chave3); //lê o primeiro do arquivo 1
    chave2=input(arq2, chave2); //lê o primeiro do arquivo 2
    chave3=input(arq3, chave1); //lê o primeiro do arquivo 3
    while (NUMERO){ //enquanto existe numero no arquivo
        if(chave1<chave2 || chave1<chave3){
            fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
            chave1=input(arq1, chave3); //lê o proximo do arquivo 1
        }else if(chave2<chave1 || chave2<chave3){
            fprintf(arqsaida, "%d\n", chave2); //escreve no arquivo saída
            chave2=input(arq2, chave2); //lê o proximo do arquivo 2
        }else if(chave3<chave1 || chave3<chave2){
            fprintf(arqsaida, "%d\n", chave3); //escreve no arquivo saída
            chave3=input(arq3, chave1); //lê o proximo do arquivo 3
        }else{
            fprintf(arqsaida, "%d\n", chave1); //escreve no arquivo saída
            chave1=input(arq1, chave3); //lê o proximo do arquivo 1
            chave2=input(arq2, chave2); //lê o proximo do arquivo 2
            chave3=input(arq3, chave1); //lê o proximo do arquivo 3
        }
    }
}

```

3. Saída dos Testes solicitados

Merging em 2 vetores

Entrada de n={2,3}

Arquivo 1

arq1.txt ×

1	1
2	2

Arquivo 2

arq2.txt ×

1	2
2	3
3	4

Saída

arqsaida.txt ×

1	1
2	2
3	3
4	4

Entrada de n={10,10}

Arquivo 1

arq1.txt ×

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Arquivo 2

arq2.txt ×

1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18
10	19

Saída

arqsaida.txt ×

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19

Merging em 3 vetores

Entrada de $n=\{10,10,10\}$

Arquivo 1

arq1.txt ×	
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Arquivo 2

arq2.txt ×	
1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18
10	19

Arquivo 3

arq3.txt ×	
1	19
2	20
3	21
4	22
5	23
6	24
7	25
8	26
9	27
10	28

Saída

arqsaida.txt ×	
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28

Matching em 2 vetores

<p>Arquivo 1</p> <pre> arq1.txt × 1 1 2 2 </pre>	<p>Entrada de $n=\{2,3\}$</p>	<p>Arquivo 2</p> <pre> arq2.txt × 1 2 2 3 3 4 </pre>
<p>Saída</p> <pre> arqsaida.txt × 1 2 </pre>		
<p>Arquivo 1</p> <pre> arq1.txt × 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 </pre>	<p>Entrada de $n=\{10,10\}$</p>	<p>Arquivo 2</p> <pre> arq2.txt × 1 10 2 11 3 12 4 13 5 14 6 15 7 16 8 17 9 18 10 19 </pre>
<p>Saída</p> <pre> arqsaida.txt × 1 10 </pre>		

Matching em 3 vetores

<p>Arquivo 1</p>	<p>Entrada de $n=\{10,10,10\}$</p> <p>Arquivo 2</p>	<p>Arquivo 3</p>
------------------	--	------------------

arq1.txt ×	
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

arq2.txt ×	
1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18
10	19

arq3.txt ×	
1	10
2	20
3	21
4	22
5	23
6	24
7	25
8	26
9	27
10	28

Saída

arqsaida.txt ×	
1	10

Merging chave dupla em 2 vetores

Arquivo 1

arq1.txt ×	
1	1
2	2

Entrada de $n=\{2,3\}$

Arquivo 2

arq2.txt ×	
1	2
2	3
3	4

Saída

arqsaida.txt ×	
1	1
2	2
3	2
4	3
5	4

Arquivo 1

Entrada de $n=\{10,10\}$

Arquivo 2

arq1.txt ×

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

arq2.txt ×

1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18
10	19

Saída

arqsaida.txt ×

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19

Merging chave duplicada em 3 vetores

Arquivo 1

Entrada de $n=\{10,10\}$
Arquivo 2

Arquivo 3

arq1.txt ×

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

arq2.txt ×

1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	17
9	18
10	19

arq3.txt ×

1	19
2	20
3	21
4	22
5	23
6	24
7	25
8	26
9	27
10	28

Saída

arqsaida.txt ×

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	10
12	11
13	12
14	13
15	14
16	15
17	16
18	17
19	18
20	19
21	19
22	20
23	21
24	22
25	23
26	24
27	25
28	26
29	27
30	28

4. Análise de solução com base nos resultados do teste de eficiência

A análise de solução foi bem complicada, pois não conseguia compreender o que o problema pedia, por cima, deu a entender que o programa deveria ler n vetores, mas depois pedia 2 vetores com 2 elementos e 3 elementos. Depois pediu 2 vetores com 10 elementos cada. E depois pediu vários vetores com quantidade máxima. Foi bem complicado de entender a especificação. Por fim, foi feito um programa que leia arquivos com os elementos dentro e assim executar. Foi bem rápida a execução.