

Nome: João Paulo de Oliveira

11611BCC046

6º Aula prática

Uberlândia

2016

## 1.Código fonte:

```
#include <stdio.h>

#include <stdlib.h>

#include "list.h"

#define NOPCOES      3

#define NCURSOS      7

#define FALSE        0

#define TRUE         1


typedef int TipoChave;


typedef struct TipoItem
{
    TipoChave Chave;

    int NotaFinal;

    int Opcao[NOPCOES];
} TipoItem;


ListElmt Registro;

List Classificacao[11];

List Aprovados[NCURSOS];

List Reprovados;


long Vagas[NCURSOS];

short Passou;

long i, Nota;
```

```
void print_list(List *list)
{

    ListElmt* element;

    void* data;

    int i;

    //fprintf(stdout, "List size is %d\n", list_size(list));

    i = 0;

    element = list_head(list);

    TipoItem *aux;

    for (i = 0; i < list_size(list); i++)
    {
        data = list_data(element);

        aux = (TipoItem*)data;

        printf("Chave: %d   Nota Final: %d\n", aux->Chave, aux->NotaFinal);

        element = list_next(element);
    }

    return;

}
```

```

void LeRegistro(ListElmt *Registro)
{
    int i;

    TipoItem *aux = malloc(sizeof(TipoItem*));

    scanf("%d %d", &aux->Chave, &aux->NotaFinal);

    for (i = 0; i < NOPCOES; i++) scanf("%d", &aux->Opcao[i]);

    Registro->data = (void*)aux;
}

int main(int argc, char *argv[])
{
    for (i = 1; i <= NCURSOS; i++) scanf("%ld", &Vagas[i-1]);

    scanf("%s*[^\\n]");

    getchar();

    for (i = 0; i <= 10; i++) list_init(&Classificacao[i], free);

    for (i = 1; i <= NCURSOS; i++) list_init(&Aprovados[i-1], free);

    list_init(&Reprovados, free);

    LeRegistro(&Registro);

    void *aux = Registro.data;

    TipoItem *aux2 = (TipoItem*)aux;

    ListElmt *element;

    while (aux2->Chave != 0)
    {

```

```

list_ins_next(&Classificacao[aux2->NotaFinal], NULL, Registro.data);

LeRegistro(&Registro);

aux = Registro.data;

aux2 = (TipoItem*)aux;
}

for (Nota = 10; Nota >= 0; Nota--)
{
    while (list_size(&Classificacao[Nota]) != 0)
    {
        element = list_head(&Classificacao[Nota]);

        if (list_size(&Classificacao[Nota]) > 1)
            list_rem_next(&Classificacao[Nota], element, &Registro.data);
        else if (list_size(&Classificacao[Nota]) == 1)
            list_rem_next(&Classificacao[Nota], NULL, &Registro.data);

        i = 1;

        Passou = FALSE;

        aux = Registro.data;

        aux2 = (TipoItem*)aux;

        while (i <= NOPCOES && !Passou)
        {
            if (Vagas[aux2->Opcao[i-1] - 1] > 0)
            {
                list_ins_next(&Aprovados[aux2->Opcao[i-1]-1], NULL, aux);
            }
        }
    }
}

```

```

        //Insere(Registro, &Aprovados[Registro.Opcao[i-1] - 1]);

        Vagas[aux2->Opcao[i-1] - 1]--;

        //Vagas[Registro.Opcao[i-1] - 1]--;

        Passou = TRUE;

    }

    i++;

}

if (!Passou) list_ins_next(&Reprovados, NULL, aux);

}

}

for (i = 1; i <= NCURSOS; i++)

{

    printf("Relacao dos aprovados no Curso%d\n", i);

    print_list(&Aprovados[i-1]);

    printf("\n");

    //Imprime(Aprovados[i - 1]);

}

printf("Relacao dos reprovados\n");

print_list(&Reprovados);

//Imprime(Reprovados);

return 0;

}

```

2.Print do funcionamento:

```
Relacao dos aprovados no Curso1
Chave: 999    Nota Final: 9
Chave: 777    Nota Final: 9

Relacao dos aprovados no Curso2
Chave: 666    Nota Final: 6

Relacao dos aprovados no Curso3
Chave: 333    Nota Final: 3

Relacao dos aprovados no Curso4
Chave: 111    Nota Final: 1

Relacao dos aprovados no Curso5

Relacao dos aprovados no Curso6

Relacao dos aprovados no Curso7

Relacao dos reprovados

Process returned 0 (0x0)   execution time : 0.088 s
Press any key to continue.
```