

		%
1	6	
2	7	
3	7	

## EINF

unidade curricular: **Algoritmia e Estruturas de Dados** ano lectivo: **2018 / 2019**

**Teste Prático 01.**      **10.Abr.2019.**      duração: **60 min** (+5 min tolerância)  
(sem consulta)

Número: \_\_\_\_\_ Nome: \_\_\_\_\_

Considere o seguinte programa em linguagem C:

```
#define DATA(node) ((node)->data)
#define NEXT(node) ((node)->next)
typedef enum _STATUS {ERROR,OK} STATUS;
typedef enum _BOOLEAN {FALSE,TRUE} BOOLEAN;
#define MAX_NOME 20

typedef struct _FUNC{
    char    nome[MAX_NOME], dept[MAX_NOME];
    float   ordenado;
    BOOLEAN do_quadro;
} FUNC;
typedef LIST_NODE * LIST;

LIST_NODE * NewNode(void *);
LIST_NODE * InsertIni(LIST *, void *);
STATUS ReadFile(LIST *,char *);

int main(int argc, char *argv[])
{
    LIST list;
    FUNC func;

    if(!ReadFile(&list,"funcionarios.txt"))
        return 0;

    FreeList(&list);
    return 1;
}

LIST_NODE *NewNode(void *data)
{
    LIST_NODE *new_node;
    if((new_node = (LIST_NODE *)malloc(sizeof(LIST_NODE))) != NULL) {
        DATA(new_node) = data;
        NEXT(new_node) = NULL;
    }
    return(new_node);
}
```

```
LIST_NODE *InsertIni(LIST *list, void *data)
{
    LIST_NODE *new_node;

    if((new_node = NewNode(data)) != NULL) {
        NEXT(new_node) = *list;
        *list = new_node;
    }
    return(new_node);
}

STATUS ReadFile(LIST *list, char *name_file)
{
    FILE *fp;
    FUNC func,*ptr;
    int aux;

    if((fp = fopen(name_file,"r")) != NULL)
    {
        while(!feof(fp)) //while(feof(fp) == 0)
        {
            fscanf(fp,"%[^;]; %[^;];%f;%d\n",func.nome, func.dept,& func.ordenado,&aux);
            if( ((ptr =(FUNC *)malloc(sizeof(FUNC)))!=NULL) && (InsertIni(list,ptr)!=NULL) )
            {
                strcpy(ptr->nome,func.nome);
                strcpy(ptr->dept,func.dept);
                ptr->ordenado = func.ordenado;
                if(aux) ptr->do_quadro = TRUE;
                else ptr->do_quadro = FALSE;
            }
            else return ERROR;
        }
        fclose(fp);
        return OK;
    }
    else return ERROR;
}

void RemoveFirstNode(LIST *list)
{
    LIST_NODE *node;

    node = NEXT(*list);
    free(DATA(*list));
    free(*list);
    *list = node;
}

void FreeList(LIST *list)
{
    while(*list != NULL)
        RemoveFirstNode(list);
}
```

**(NOTA: todas as funções desenvolvidas devem ser invocadas a partir do main)**

**01** Desenvolva o código necessário para mostrar no ecrã o valor total dos ordenados pagos aos FUNC da lista.

**02** Desenvolva o código necessário para passar para o quadro um FUNC cujo nome deve ser lido através do teclado (passar de *do\_quadro* = FALSE para *do\_quadro* = TRUE).

---

**EINF. Teste Prático 01. 11.Abr.2018.**

unidade curricular: **Algoritmia e Estruturas de Dados**    ano lectivo: **2018 / 2019**

---

- 03** Desenvolva o código necessário para eliminar da lista todos os FUNC que não pertençam ao quadro (*do\_quadro* = FALSE).