

Q3:

Sabendo-se que uma estrutura de impressão com prioridade representada por:

```
typedef struct prior{
    int p_user, id;
    char narq[31];
}TP;

typedef struct ei{
    int tam, ult_pos_ocup;
    TP **vet;
}TEI;
```

E as seguintes operações:

```
TEI * inicializa(int tam){
    TEI *resp = (TEI *) malloc(sizeof(TEI));
    resp->vet = (TP **) malloc(sizeof(TP *) * tam);
    int i;
    for(i = 0; i < tam; i++) resp->vet[i] = NULL;
    resp->tam = tam;
    resp->ult_pos_ocup = -1;
    return resp;
}

void imprime(TEI *resp){
    int i;
    for(i = 0; i <= resp->ult_pos_ocup; i++)
        printf("%d:  %d  %s\n",  resp->vet[i]->p_user,  resp->vet[i]->id,
resp->vet[i]->narq);
}

void retira(TEI *resp){
    if(resp->ult_pos_ocup == -1) return;
    TP *temp = resp->vet[0];
    int i;
    for(i = 0; i < resp->ult_pos_ocup; i++)
        resp->vet[i] = resp->vet[i + 1];
    resp->ult_pos_ocup--;
    free(temp);
}
```

Escreva a função de inserção nessa estrutura, se houver espaço nela. Essa função recebe a estrutura (**TEI *resp**), a prioridade do usuário (campo **p_user**), o identificador do processo (campo **id**) e o nome do arquivo (campo **narq**), e insere de acordo com os seguintes passos: (a) aloca espaço para esse novo elemento, e (b) esse novo elemento de impressão será inserido depois do último elemento com **p_user** igual ao seu. Isto quer dizer que, em alguns casos, você terá que remanejar elementos. O protótipo da função é o seguinte: **void insere(TEI *resp, int p_user, int id, char *narq);**