**RESPOSTAS P1 – ESTRUTURA DE DADOS**

1. **Inserção de um elemento (push) – PILHA DINÂMICA**

bool inserirElemPilha(PILHA\* p, REGISTRO reg) {

PONT novo = (PONT) malloc(sizeof(ELEMENTO));

novo->reg = reg;

novo->prox = p->topo;

p->topo = novo;

return true;

}

1. **Exclusão de um elemento – FILA DINÂMICA**

bool excluirDaFila(FILA\* f, REGISTRO\* reg) {

if (f->inicio==NULL) return false; if (f->nroElem==0)

\*reg = f->inicio->reg; // \*reg = f->A[f->inicio];

PONT apagar = f->inicio; // \*reg = f->A[f->inicio];

f->inicio = f->inicio->prox;

free(apagar); // f->nroElem--;

if (f->inicio == NULL) f->fim = NULL;

return true;

}

1. **Exibição/Impressão – LISTA LIGADA DINÂMICA**

void exibirLista(LISTA\* l){

PONT end = l->inicio;

printf("Lista: \" ");

while (end != NULL) {

printf("%i ", end->reg.chave);

end = end->prox;

}

printf("\"\n");

}

1. **Busca binária – LISTA LINEAR SEQUENCIAL ESTÁTICA**

int buscaBinaria(LISTA\* l, TIPOCHAVE ch) {

int esq, dir, meio;

esq = 0;

dir = l->nroElem-1;

while(esq <= dir) {

meio = ((esq + dir) / 2);

if(l->A[meio].chave == ch) return meio;

else {

if(l->A[meio].chave < ch) esq = meio + 1;

else dir = meio - 1;

}

}

return -1;

}