

```
#include <stdio.h>
#include <stdlib.h>

#define true 1
#define false 0

typedef int bool;

/* Implementacao gulosa para a solucao do problema da alocao de
atividades. */
int selecaoGulosa(int ini[], int fim[], int n) {
    int i, ultimaSelecionada = 0;
    int selecionadas = 0;
    if (n==0) return 0;
    printf("a1 ");
    selecionadas++;
    for (i=1; i<n; i++)
        if (ini[i]>=fim[ultimaSelecionada]) {
            printf("a%i ", i+1);
            selecionadas++;
            ultimaSelecionada = i;
        }
    printf("\n");
    return selecionadas;
}
```

```

int* selAtual;
int* selMelhor;
int contAtual;
int contMelhor;

bool ehViavel(int ini[], int fim[], int atual){
    int i;
    for(i=0;i<contAtual;i++)
        if ((ini[atual]>=ini[selAtual[i]] && ini[atual]<fim[selAtual[i]])
|| (fim[atual]>ini[selAtual[i]] && fim[atual]<=fim[selAtual[i]]) ||
        (ini[atual]<=ini[selAtual[i]] && fim[atual]>=fim[selAtual[i]]))
            return 0;
    return 1;
}

void selecaoTentativaEErroAux(int ini[],int fim[],int n,int atual) {
    int i;
    if(atual == n){
        if (contAtual > contMelhor){
            contMelhor = contAtual;
            for(i=0;i<contMelhor;i++){
                selMelhor[i] = selAtual[i];
            }
        }
        return;
    }

    selecaoTentativaEErroAux(ini, fim, n, atual+1);
    if (ehViavel(ini, fim, atual)){
        selAtual[contAtual] = atual;
        contAtual++;
        selecaoTentativaEErroAux(ini, fim, n, atual+1);
        selAtual[contAtual] = -1;
        contAtual--;
    }
}

int selecaoTentativaEErro(int ini[], int fim[], int n) {
    int i;
    selAtual = (int*)malloc(sizeof(int)*n);
    selMelhor = (int*)malloc(sizeof(int)*n);
    contAtual = 0;
    contMelhor = 0;
    selecaoTentativaEErroAux(ini, fim, n, 0);
    for (i=0;i<contMelhor;i++)
        printf("a%i ", selMelhor[i]+1);
    printf("\n");
    free(selAtual);
    free(selMelhor);
    return contMelhor;
}

```

```

/* Funcao main que cria uma instancia do problema (horario das
atividades) e
   chama a funcao gulosa e usando tentativa e erro para resolver.
   As atividades estao ordenadas de acordo com seu horario de
termino.          */
int main() {
    int inicio[] = {1,3,0,5,3, 5, 6, 8, 8, 2,12};
    int fim[] =    {4,6,7,8,9,10,11,12,13,14,15};

    printf("Executando Algoritmo Guloso.\n");
    int total = selecaoGulosa(inicio,fim,11);

    printf("Foram selecionadas %i atividades.\n", total);

    printf("\nExecutando algoritmo usando Tentativa e Erro.\n");
    total = selecaoTentativaEErro(inicio,fim,11);

    printf("Foram selecionadas %i atividades.\n", total);

    return 0;
}

/* SAIDA

Executando Algoritmo Guloso.
a1 a4 a8 a11
Foram selecionadas 4 atividades.

Executando algoritmo usando Tentativa e Erro.
a1 a4 a8 a11
Foram selecionadas 4 atividades.

*/

```