

```
/* Gabarito P2 - INF1620 - 2008.2
```

```
*/
```

```
/* Questao 1
```

```
*/
```

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

```
struct aeroporto
{
    char sigla[4];
    char cidade[51];
    char pais[31];
    float taxa;
    int capacidade;
};
typedef struct aeroporto Aeroporto;
```

```
Aeroporto* Cadastra(char* s, char* c, char* p, float tx, int cap)
```

```
{
    Aeroporto *a = (Aeroporto*) malloc(sizeof(Aeroporto));
    if (a == NULL) return NULL;
    strcpy(a->sigla, s);
    strcpy(a->cidade, c);
    strcpy(a->pais, p);
    a->taxa = tx;
    a->capacidade = cap;

    return a;
}
```

```
int main(void)
```

```
{
    Aeroporto *aero=Cadastra("SKG", "Thessaloniki", "Grecia", 73.98, 20);

    printf("%s\n",aero->sigla);
    printf("%s\n",aero->cidade);
    printf("%s\n",aero->pais);
    printf("%f\n",aero->taxa);
    printf("%d\n",aero->capacidade);

    return 0;
}
```

```
/* Questao 2
```

```
*/
```

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

```
float media(float* despesas, int m, int n, int i, int j, int k)
```

```
{
    int x;
    float y;
    for (x=j;x<=k;x++)
        y = y + despesas[n*i+x];
    y = y/(k-j+1);
    return y;
}
```

```
int main(void)
```

```
{
    float m;
    float matriz[20] = {5.65, 7.54, 5.41, 4.86, 6.24,
                        145.45, 134.52, 124.52, 135.63, 159.86,
```

```
49.85, 54.88, 63.87, 65.78, 78.35,  
81.29, 89.62, 75.64, 87.89, 77.55};
```

```
m=media(matriz, 4, 5, 1, 2, 4);
```

```
printf("%f\n",m);
```

```
return 0;
```

```
}
```

```
/* Questao 3
```

```
*/
```

```
static int comp_eventos (const void* p1, const void* p2)
```

```
{
```

```
Evento **e1 = (Evento**)p1;
```

```
Evento **e2 = (Evento**)p2;
```

```
if ((*e1)->inicio.mes < (*e2)->inicio.mes) return -1;
```

```
if ((*e1)->inicio.mes > (*e2)->inicio.mes) return 1;
```

```
if ((*e1)->inicio.dia < (*e2)->inicio.dia) return -1;
```

```
if ((*e1)->inicio.dia > (*e2)->inicio.dia) return 1;
```

```
if ((*e1)->inicio.hora < (*e2)->inicio.hora) return -1;
```

```
if ((*e1)->inicio.hora > (*e2)->inicio.hora) return 1;
```

```
if ((*e1)->inicio.minuto < (*e2)->inicio.minuto) return -1;
```

```
return 1;
```

```
}
```

```
void Cronologica(Evento** agenda, int n)
```

```
{
```

```
qsort(agenda, n, sizeof(Evento*), comp_eventos);
```

```
}
```