

MC 102Z – Segunda prova - 18/10/12

1. Escreva a função correspondente ao protótipo da função `calc()`, definido a seguir. Essa função tem como parâmetro um vetor de valores `float`. A execução da função deve determinar os valores máximo, mínimo desse vetor e também a média aritmética. Esses três valores devem ser retornados pela função como parâmetros passados por referência. O sinônimo `NP` define o número de elementos do vetor.

```
#define NP 8
void calc(float notas[], float *max, float *min, float *med);

void calc(float notas[], float *max, float *min, float *med){
    int i;
    float s = notas[0];
    *max = notas[0];
    *min = notas[0];
    for(i = 1; i < NP; i++){
        if(notas[i] > *max) *max = notas[i];
        if(notas[i] < *min) *min = notas[i];
        s += notas[i];
    }
    *med = s / NP;
}
```

2. Usando as declarações feitas a seguir, das variáveis `rep`, `apr` e `ex`, da estrutura `Aluno`, escreva a função `defSituacao()` cujo protótipo é mostrado a seguir. Essa função deve calcular a média das notas de cada aluno através da função `calc()` cuja descrição e protótipo foram apresentados na questão anterior e definir a situação do aluno segundo o seguinte critério de aprovação: se a menor das notas for menor que 3.0 e a maior das notas for menor que 8.0 o aluno é reprovado, sem direito a exame. Caso contrário (menor nota maior ou igual a 3.0 ou maior nota maior ou igual a 8.0) se a média for menor que 5.0, o aluno deve fazer exame, senão o aluno é aprovado.

```
char * rep = "reprovado";
char * apr = "aprovado";
char * ex = "exame";

struct Aluno{
    int ra;
    char *nome;
    int curso;
    float notas[NP];
    char *situacao;
    float mf;
};

void defSituacao(struct Aluno *p);

void defSituacao(struct Aluno *p){
    float max, min;
    calc((*p).notas, &max, &min, &(*p).mf);
    if((min < 3) && (max < 8)) (*p).situacao = rep;
    else if((*p).mf < 5) (*p).situacao = ex;
        else (*p).situacao = apr;
}
```

3. Usando um vetor de estruturas do tipo `Aluno` representando uma turma, escreva uma função que imprime a menor nota de toda a turma, a maior nota de toda a turma e a média geral da turma. A função deve obedecer ao seguinte protótipo:

```
void result(struct Aluno turma[]);
```

```

void result(struct Aluno turma[]){
    float max,tmax, min,tmin, med,tmed;
    int i;
    calc(turma[0].notas,&tmax,&tmin,&tmed);
    for(i = 0; i < NA; i++){
        calc(turma[i].notas,&max,&min,&med);
        if(max > tmax) tmax = max;
        if(min < tmin) tmin = min;
        tmed += med;
    }
    printf("maior nota:%5.2f menor nota:%5.2f media geral:%5.2f\n",
        tmax,tmin, (tmed/NA));
}

```

4. Escreva a função definida pelo protótipo mostrado abaixo para buscar por um aluno numa turma a partir do seu nome, passado como parâmetro. Para implementar essa função pode ser usada a função strcmp(), definida em <strings.h> e usada na última atividade de laboratório. Essa função deve retornar o índice do aluno procurado ou -1 no caso de o aluno não ser encontrado.

```

int buscaPorNome(struct Aluno turma[], char *nome);

int buscaPorNome(struct Aluno turma[], char *nome){
    int i;
    for(i = 0; i < NA; i++)
        if(strcmp(turma[i].nome,nome) == 0) return i;
    return -1;
}

```

Observações:

- As questões têm o mesmo valor
- Nas questões 3. e 4. Suponha o sinônimo NA para indicar o número de alunos da turma.
- No verso é mostrado um programa exemplo que usa algumas das funções pedidas no enunciado.

```

#define NP 8
#define NA 7

void calc(float notas[], float *max, float *min, float *med);

char * rep = "reprovado";
char * apr = "aprovado";
char * ex = "exame";

struct Aluno{
    int ra;
    char *nome;
    int curso;
    float notas[NP];
    char *situacao;
    float mf;
};

void defSituacao(struct Aluno *p);

void escreve(struct Aluno *p){
    float max, min, med;
    calc((*p).notas, &max, &min, &med);
    printf(
        "ra:%d\t nome:%s\t mef:%5.2f situacao:%s\n",
        (*p).ra, (*p).nome, (*p).mf, (*p).situacao
    );
}

struct Aluno turmaZ[] = {
    {1, "Huguinho", 32, { 5, 5, 6, 7, 8, 7, 6, 3 } },
    {2, "Zezinho", 32, { 5, 4, 4, 4, 5, 4, 5, 4 } },
    {3, "Luizinho", 32, { 6, 6, 7, 7, 7, 6, 7, 8 } },
    {4, "Donald", 32, { 4, 5, 3, 5, 2, 5, 6, 4 } },
    {5, "Mickey", 34, { 7, 7, 7, 8, 7, 8, 8, 7 } },
    {6, "Minnie", 34, { 8, 9, 8, 9, 9, 8, 9, 9 } },
    {7, "Margarida", 34, { 9, 8, 9, 9, 9, 8, 7, 9 } }
};

int main(){
    int i = 0;
    for(i = 0; i < NA; i++) {
        defSituacao(&(turmaZ[i]));
        escreve(&(turmaZ[i]));
    }
}

```

- A função strcmp() é definida de acordo com o protótipo mostrado abaixo e compara os strings passados como parâmetro, retornando o valor zero se os dois strings forem iguais.

```
int strcmp(char *s1, char *s2);
```