

MC102 - Algoritmos e Progração de Computador

Prof. Alexandre Xavier Falcão

15° Aula: Registros

1 Registros

Sabemos que variáveis compostas são aquelas que agrupam um certo número de elementos em um único identificador. No caso de vetores e matrizes, todos os elementos agrupados são do mesmo tipo e, portanto, dizemos que são variáveis compostas homogêneas.

Em muitas situações, porém, desejamos agrupar variáveis de tipos diferentes. Um exemplo é o caso em que agrupamos dados sobre uma pessoa/objeto (e.g. RA, nome, e notas de um aluno). O conceito de registro permite criar um identificador único associado a todos esses dados. Portanto, **um registro é uma variável composta heterogênea** e seus elementos são chamados **campos**.

Diferentes tipos de registro podem ser criados com campos diferentes. Para definir um tipo de registro, nós usamos o comando **typedef struct**. O programa abaixo define um tipo de registro, Aluno, uma variável deste tipo, armazena dados nos seus campos e depois imprime os dados armazenados.

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

typedef struct _aluno {
    int    RA;
    char  nome[50];
    float nota[3];
} Aluno;

int main()
{
    Aluno a; /* variável do tipo Aluno */

    a.RA      = 909090;
    strcpy(a.nome, "Jose Maria");
    a.nota[0] = 3.0;
    a.nota[1] = 10.0;
    a.nota[2] = 7.0;
    printf("%6d %s %5.2f %5.2f %5.2f\n", a.RA, a.nome, a.nota[0], a.nota[1], a.nota[2]);
    return 0;
}
```

Observe que cada campo do registro é uma variável de qualquer tipo válido, incluindo um **outro registro, vetor, matriz**, etc.

```

#include <stdio.h>

typedef struct _ponto {
    float x;
    float y;
} Ponto;

typedef struct _reta {
    Ponto p1;
    Ponto p2;
} Reta;

typedef struct _curva { /* pontos consecutivos são interligados
                        por segmentos de reta. */
    Ponto pt[100];
    int npts;
} Curva;

int main()
{
    Reta r;
    Curva c;
    int i;

    /* ler os pontos da reta */

    scanf("%f %f",&r.p1.x,&r.p1.y);
    scanf("%f %f",&r.p2.x,&r.p2.y);

    /* ler os pontos da curva */

    scanf("%d",&c.npts);
    for (i=0; i < c.npts; i++)
        scanf("%f %f",&c.pt[i].x,&c.pt[i].y);

    /* complete o programa para que ele verifique se existe
       intersecção entre a curva e a reta. */

    return 0;
}

```

Vetores de registros podem ser usados para armazenar base de dados (ou parte da base) em memória. O programa abaixo ilustra o armazenamento de uma mini base com 5 nomes e 5 telefones.

```

#include <stdio.h>

```

```

typedef struct _agenda {
    char nome[50];
    int telefone;
} Agenda;

int main()
{
    Agenda amigo[5];
    char nome_aux[50];
    int i, comp;

    printf("Entre com os nomes\n");
    for (i=0; i < 100; i++) {
        fgets(nome_aux, 49, stdin);
        comp = strlen(nome_aux);
        strncpy(amigo[i].nome, nome_aux, comp-1); /* elimina \n */
        amigo[i].nome[comp-1] = '\0'; /* insere \0 por garantia */
    }

    printf("Entre com os telefones\n");
    for (i=0; i < 5; i++)
        fscanf(stdin, "%d", &amigo[i].telefone);

    for (i=0; i < 5; i++)
        fprintf(stdout, "%s: %d\n", amigo[i].nome, amigo[i].telefone);

    return 0;
}

```

2 Exercícios

1. Complete o programa acima para cálculo de intersecções entre a reta e a curva.
2. O centro de gravidade (x_g, y_g) de um conjunto de pontos (x_i, y_i) , $i = 0, 1, \dots, n - 1$, é definido por:

$$\begin{aligned}
 x_g &= \frac{\sum_{i=0}^{i=n-1} x_i}{n} \\
 y_g &= \frac{\sum_{i=0}^{i=n-1} y_i}{n}.
 \end{aligned}$$

Faça um programa para ler os pontos de uma curva e calcular seu centro de gravidade.

3. Faça um programa para armazenar 20 nomes e 20 telefones em um vetor de registros, colocar os elementos do vetor em ordem crescente de nome, e depois imprimir o vetor ordenado.