Algoritmos e Estruturas de Dados
DWM(1º ano)

Ficha 6

Ano Lectivo de 2021/22

1 Definição de Tipos

1. Para gerir a informação sobre os alunos inscritos a uma dada disciplina, é necessário armazenar os seguintes dados:

   - Nome do aluno(string com no máximo 100 caracteres)
   - Número do aluno
   - Avaliação

   Quanto à avaliação, existem dois métodos alternativos:

   - No método A, a avaliação tem uma componente periódica (6 mini-testes quinzenais), e uma componente final (teste e/ou exame).
   - No método B, a avaliação consta apenas do teste e/ou exame final.

   Considere que para cada prova, os resultados são armazenados com um número inteiro (de 0 a 100).

   (a) Defina os tipos **Avaliacao**, **Aluno** e **Turma**. Assuma que o número de alunos nunca ultrapassa 100, podento por isso usar um array para armazenar a informação da turma.

   (b) Defina uma funcão **int acrestentaAluno(Turma t, Aluno a)** que acrescenta a informação dum dado aluno a uma turma. A função deverá retornar 0 se a operação for feita com sucesso.

   (c) Defina uma função **int procura(Turma t, int numero)** que procura esse aluno na turma. A função deverá retornar -1 se a informação desse aluno não existir; caso exista deve retornar o índice onde essa informação se encontra.

   (d) Defina uma função que calcula a nota final (um float de 0 a 20) de um aluno.

   - Para alunos do método A, a nota final é a média ponderada da parte periódica(40%) com a nota do teste/exame (60%).
   - A nota da parte períodica é obtida como a média das 5 melhores notas dos mini testes.
   - A nota do teste/exame é a nota do teste caso o aluno não tenha realizado o exame; é a nota do exame no outro caso.

   (e) Defina uma função que determine quantos alunos obtiveram aproveitamento à disciplina (nota final maior ou igual a 10).

2. No plano cartesiano um rectângulo com os lados paralelos aos eixos pode ser univocamente determinado por uma diagonal dada por dois pontos. Assim para representar esta figura geométrica definiu-se os seguintes tipos de dados:

```
typedef struct sPonto{
    float x;
    float y;
} Ponto;

typedef struct sRectangulo{
    Ponto p1;
    Ponto p2;
} Rectangulo;
```

Especifique as seguintes funções e utilize-as num programa.

```
float area( Rectangulo r){
    ...
}

float Perimetro( Rectangulo r){
    ...
}
```

3. Pretende-se guardar a informação sobre os resultados dos jogos duma jornada dum campeonato de futebol na seguint estrutura de dados:

```
typedef int Golos;

typedef char Equipa[300];

typedef struct sInterv{
    Equipa e;
    Golos g;
} Interv;

typedef struct sJogo{
    Interv i1;
    Interv i2;
} Jogo;

typedef Jogo Jornada[20];
```

Desta forma podemos ter um programa que vai trabalhar as jornadas e que declara as seguintes variáveis:

```
int main(int argc, char* argv[]){
    Jornada j1, j2;
    Jornada campeonato [56];
    ...
}
```

Depois de analisar bem a estrutura de dados especifique as seguintes funções e crie um programa para as testar.

```
int igualj(Jornada j); //que verifica se nenhuma equipa joga com ela própria
int semrepet(Jornada j); // que verifica se nenhuma equipa joga mais do que um jogo
??? empates( Jornada j); // que dá a lista dos pares de equipas que empataram na jornada
??? equipas( Jornada j); //que dá a lista das equipas que participam na jornada
??? calcres(Jornada j); // que calula os pontos que cada equipa obteve na jornada (venceu
```

Como exercício extra defina tipos de dados para suportar o tipo de resultado de algumas funções (aquelas qie têm o tipo a ???).

2 Listas Ligadas

1. Considere uma lista de inteiros (não se sabe o seu comprimento). Especifique então as seguintes funções e estruturas de dados:

   (a) Defina os tipos necessários para suportar uma lista ligada de inteiros.

   (b) Especifique uma função para inserir um valor na cabeça da lista

   (c) Especifique uma função para listar os valores da lista, do ínicio para o fim (faça também a função que lista os elementos na ordem inversa).

   (d) Especifique uma função para procurar um valor na lista (como resultado deverá devolver um apontador para o elemento ou NULL caso não o encontre).

   (e) Especifique uma função para contar os elementos da lista.

   (f) Especifique uma função para calcular o maior elemento da lista.

   (g) Especifique um programa, usando as funções definidas, que crie uma lista com os múltiplos de 3 entre 0 e 100 e os lista por ordem descrescente e crescente.

2. Pretende-se que desenvolva uma aplicação para gerir uma agenda de contactos. Uma agenda é uma lista de entradas ou grupo de entradas. Uma entrada tem a seguinte constituição:

   - chave chave única de identificação (não pode haver duas entradas com a mesma chave);
   - tipo tipo de entrada: pessoa, empresa, ...
   - nome nome da pessoa, empresa ou entidade;
   - email contacto electrónico (é opcional)
   - telefone numero de telefone (é obrigatório)

   Um grupo pode ter entradas, referências a entradas já existentes na agenda (por chave) ou subgrupos (os grupos podem ter grupos aninhados infinitamente). O grupo tem, então, a seguinte constituição:

   - chave chave única de identificação (não pode haver dois grupos com a mesma chave);
   - nome nome do grupo; lista de itens; entradas e/ou grupos e/ou referências;

   Por sua vez, a referência é apenas constituída pela chave de entrada ou grupo de referência,

   Desenvolva a aplicação nas seguintes etapas:

   (a) Defina as estruturas de dados necessárias para suportar o sistema de informação,

   (b) Especifique as várias funções de inserção: inserir uma entrada na agenda, inserir um grupo na agenda, inserir uma entrada num grupo, ...

(c) Especifique uma função para listar o conteúdo da agenda

(d) Especifique uma função para gravar o conteúdo da agenda num ficheiro

(e) Especifique uma função para carregar o conteúdo da agenda dum ficheiro.

(f) Especifique as várias funções de remoção.