UNICAP – CCT – Ciência da Computação
Estrutura de Dados II – INF 1217 – Noite
Prof. Márcio Bueno
2º GQ – Junho/2009 – Aluno: _____

Não é permitida a desistência após o aluno ter acesso à prova.
O aluno deverá esperar pelo menos 30 minutos para entregar a prova.
Só serão consideradas as respostas que estiverem na folha pautada.
Algoritmos sem endentação ou utilizando variáveis globais serão desconsiderados.

Questão 1 (1,5 ponto) Considerando uma função de hashing h(x) = x % 5 e utilizando encadeamento interno com área de sinônimos com 5 espaços para o tratamento de colisões, mostre a hashing resultante após a inserção dos seguintes valores (deixe os cálculos junto):

1. Dia do seu nascimento
2. Mês do seu nascimento
3. Ano do seu nascimento (dois dígitos)
4. A sua idade
5. O valor 5 for homem e o valor 7 se for mulher
6. A quantidade de letras do seu primeiro nome
7. A quantidade de letras do seu último sobrenome

Obs: não é permitido inserir valores repetidos.

Questão 2 (2 pontos) Implemente a função de hashing da Questão 1, assim como a função de inserção na hashing. **Obs:** não é permitido inserir valores repetidos.

Questão 3 (1,5 ponto) Realize o passo a passo de ordenar o vetor {15, 11, 7, 19, 25, 20, 4, 8, 3, 23, 1 ,17, 21, 5, 14} utilizando MergeSort.

Questão 4 (2 pontos) Implemente o ShellSort que recebe duas funções de ordenação como parâmetro. Para que o ShellSort funcione é preciso calcular os incrementos através do método visto em sala (**dica**: $3*h_i + 1$). Não precisa implementar as funções que foram recebidas como parâmetro que possuem a seguinte assinatua:

```
void funcaoDeOrdenacao(int v[], int n, int h, int np);
```

h é o incremento e **np** é partição selecionada dentre as possíveis para um determinado valor de **h**, logo o intervalo válido para **np** é $0 \leq np < h$.

Questão 5 (3,0 pontos, válido para SUBSTITUIR a nota do trabalho) Implemente o algoritmo QuickSort assumindo que o pivô está no meio do vetor e que você deve utilizar dois apontadores para percorrer o vetor e realizar as trocas para as criações das partições, um apontador começa no início do vetor e será incrementado até achar um valor maior que o pivô, enquanto o outro apontador começa no fim do vetor e será decrementado até achar um valor menor que o pivô, e nesse momento é realizada a troca dos dois valores apontados por estes apontadores. Veja o momento de parada. A função quicksort pode ser recursiva, porém a função partição obrigatoriamente deve ser iterativa. **Obs:** se você implementou esta questão a nota do seu trabalho será DESCONSIDERADA, mesmo que tenha sido maior.

Boa Prova!