Comp Sci 1MD3

Mid-Term II 2003 *answers*

Dr. Jacques Carette

Name: ______________________________

Student No.: ___________________________

Duration : 50 minutes

- This midterm contains 19 questions on 4 double-sided pages (including this one).
- This midterm will be marked out of 50. There are 55 total marks available.
- Answer the question in the space provided.
- Do not separate the pages.
- Make sure that you do not get stuck on one question; use your time wisely.

---

1. (Basic functional programming) Assume that the following Maple statements have already been executed

   ```
   L := [1,2,3,4,5,6,7];
   g := x -> (y->x*y);
   h := [[3], [3, 9], [3, 9, 27], [3, 9, 27, 81], [3, 9, 27, 81, 243]];
   ```

   What is the result of executing the following Maple statements (1 point each): *[3]*

   (a) map($x \rightarrow 10 * x$,L);

   (b) map(g(a), L);

   (c) map($x \rightarrow \text{add}(i, i = x)$,h);

   (a) [10,20,30,40,50,60,70]

   (b) [a,2*a,3*a,4*a,5*a,6*a,7*a]

   (c) [3,12,39,120,363]

2. All of the following questions are true/false and worth 1 point each. *[8]*

   (a) Black box testing allows you to look at the code to generate tests. *false*

   (b) Coverage testing tells you if every execution path in a routine is tested. *false*

   (c) It is possible to have a C function return a function pointer as a return value. *true*

   (d) XML is a programming language. *false*

   (e) XSL is a programming language. *true*

   (f) It is possible to implement a linked-list of bounded length using an array. *true*

   (g) A skip list is a variation on a linked-list with a (random) number of forward pointers. *true*

   (h) The “relations” that one finds in a database are different than relations in mathematics. *false*

3. Name the programming paradigm of each the following languages (1 point each): *[4]*

   (a) Java - *OO (object oriented)*

   (b) C - *imperative*

   (c) Cobol - *imperative*

   (d) Structured Query Language (SQL) - *declarative*

4. SVG and postscript use the same technique to represent, respectively, 2D graphics and printer instructions. What technique do they use? *[1]*
   represent data as programs

5. Show how the array below would be arranged in main memory when stored in row major order. *[2]*

5	3	7
4	2	8
1	9	6

5	3	7	4	2	8	1	9	6

6. Which one of these would be a good reason to use a linked list instead of an array to store large amounts of data? *[2]*

   (a) the data is ordered

   (b) the data is dynamic (many inserts and deletes)

   (c) there are few searches of the data

   (d) *all of the above*

   (e) none of the above

7. Which of the following has the most space overhead to store ordered data? [overhead is defined as space that is not used to store data but still needed by the datastructure] *[2]*

   (a) *an array*

   (b) a linked list [this turned out to be a badly phrased question!]

   (c) *a doubly-linked list*

   (d) *a binary tree*

8. Which of the following datastructures has the drawback of "creeping forward" in memory during normal use ? *[2]*

   (a) an array

   (b) *a queue*

   (c) a circular stack

   (d) a doubly linked list

9. Which of the following methods to resolve collisions in a hash table result in primary clustering? *[2]*

   (a) double hashing

   (b) extend buckets with a linked list

   (c) *put the elements in the next available slot*

   (d) put the elements in the next available slot, looking at slots $(h(k) + i^2) \bmod m$

10. The structure definition

```
struct foo {
    int i;
    struct foo *a;
    struct foo *b;
}
```

can be natural for implementing two different datastructures. Which are they? *[2]*

   (a) a doubly linked list and a linked list

   (b) a queue and a stack

   (c) a circular queue and a doubly linked list

   (d) *a doubly linked list and a tree*

11. Given a hash table of size m where collisions resolution is done by extending hash buckets with linked list, what does the statement ”this hash table currently has a load factor of 2” mean? *[2]*

   (a) there are 2 elements stored in the hash table

   (b) only two more elements can be stored in the hash table

   (c) all the linked lists have length 2

   (d) *there are* $2 * m$ *elements in the table*

12. Which one of these is not the main function of the kernel of an Operating System? *[2]*

   (a) *Manage windows*

   (b) Manage running programs

   (c) Manage hardware

   (d) Manage files

13. Which of the following is not a network topology? *[2]*

   (a) Star

   (b) *Doughnut*

   (c) Ring

   (d) Bus

   (e) Irregular

14. What does IP stand for in the context of networks? *[2]*

   (a) Intellectual Property

   (b) *Internet Protocol*

   (c) Internet Property

   (d) Intellectual Protocol

15. What is the result of the following computations in IEEE standard floating point arithmetic? (1 point each) *[4]*

    (a) -1/0. = -Float(∞)

    (b) Float(∞)+12 = Float(∞)

    (c) -0./0. = NaN (or Float(undefined))

    (d) 1.0e40 + 1.0 = 1.0e40

16. Rewrite the print() function in the following code to print a tree of characters, stored alphabetically, in reverse order. *[4]*

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

/* use char as datatype */
typedef char datatype;

/* useful short-hand */
#define empty_tree (tree *)NULL

/* actual definition of a binary tree type */
typedef struct tree {
    datatype data;
    struct tree *left;
    struct tree *right;
} tree;

void print(tree *t) {
    if (t!=empty_tree) {
        print1(t->left);
        printf("%c ", t->data);
        print1(t->right);
    }
}

/* new version */
void print(tree *t) {
    if (t!=empty_tree) {
        print1(t->right);
        printf("%c ", t->data);
        print1(t->left);
    }
}
```

17. Rewrite the function g from the following C fragment to use the *pifi* table. The body of your new function should consist of exactly 1 line. *[3]*

```
extern int f1(int);
extern int f2(int);
extern int f3(int);
extern int f4(int);

typedef int (*pifi)(int);

static pifi tab[7] = {f2, f3, f1, f1, f1, f4, f2};

int g(int i) {
    int res;

    switch (i%7) {
    case 0: res = f2(i*i); break;
    case 1: res = f3(i*i); break;
    case 2: res = f1(i*i); break;
    case 3: res = f1(i*i); break;
    case 4: res = f1(i*i); break;
    case 5: res = f4(i*i); break;
    case 6: res = f2(i*i); break;
    }
    return res;
}

/* new version */
int g(int t) {
    return tab[i%7](i*i);
}
```

18. Rewrite the following code (which computes the factorial function) to be recursive (and not use any form of loop). *[4]*

```
long f(long in) {
    long i,acc;
    for (acc=1,i=1; i<=in; i++)
        acc *= i;
    return acc;
}

/* new version */
long f(long in) {
    if (in==0)
        return 1.0;
    else
        return in*f(in-1);
}
```

19. The table below represents the contents of some cells in a computer's main memory along with the address of each cell represented. Note that some of the cells contain letters of the alphabet, and each such cell is followed by an empty cell. Place addresses in these empty cells so that each cell containing a letter together with the following cell form an entry in a linked list in which the letters appear in alphabetical order. (Use zero for the NULL pointer.) What address should the head pointer contain? *[4]*

Address	Contents
11	C
12	15
13	G
14	19
15	E
16	21
17	B
18	11
19	U
20	0
21	F
22	13

The head pointer should contain address 17

END TEST