TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2007

MIDTERM TEST WITH ANSWERS

COMP 206
PROGRAM AND DATA STRUCTURES

WITH ANSWERS

Time Allowed: 90 minutes

Instructions: There are five questions:

- Question 1 is worth 18 marks
- Question 2 is worth 12 marks
- Question 3 is worth 16 marks
- Question 4 is worth 20 marks
- Question 5 is worth 24 marks

The exam will be marked out of 90.
You should attempt all the questions.
Write your answers in the exam paper and hand in all sheets.
Operator precedences are given in Appendix A.
There are spare pages at the end of the book for extra working, if required.

Student ID: ....

Question 1. [18 marks]

(a) [12 marks]

In the box below give the output of the following program:

```
#include <stdio.h>

int main()
{
   int c = 4;
   int *p;
   char a[] = "COMP 206";
   float matrix[2][3] = {{1.0, 2.0, 4.0}, {8.0, 16.0, 32.0}};

printf("a = %s\n",  a);
printf("*a + 3 = %c\n", *a + 3);
printf("*(&c) = %d\n", *(&c));
printf("matrix[1][1]= %f\n", matrix[1][1]);
printf("*matrix[1] + 1 = %f\n", *matrix[1] + 1);
printf("(*matrix)[0] = %f\n", (*matrix)[0]);

   return 0;
}
```

```
a = COMP 206
*a + 3 = F
*(&c) = 4
matrix[1][1]= 16.000000
*matrix[1] + 1 = 9.000000
(*matrix)[0] = 1.000000
```

2 marks for each

(Question 1 continued on next page)

(Question 1 continued)

(b) [6 marks] You want to write a program which will take the name of a file as an argument and write it to the terminal. Suppose that the file `art1.txt` has the English text of Article 1 of the Treaty of Waitangi. Then you would get this behaviour:

```
%gcc moreorless.c -o moreorless
%./moreorless art1.txt
The Chiefs of the Confederation of the United Tribes of New Zealand and the
separate and independent Chiefs who have not become members of the Confederation
cede to Her Majesty the Queen of England absolutely and without reservation all
the rights and powers of Sovereignty which the said Confederation or Individual
Chiefs respectively exercise or possess, or may be supposed to exercise or to
possess over their respective Territories as the sole sovereigns thereof.
```

You choose to implement the program by:

- opening a file;
- calling a function, `copyout`, to copy the file to the terminal;
- closing the file.

Complete the boxes in the following code:

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

```
/* prototype of copyout function */
void copyout(FILE *ifp);
```

2 marks: correct types is important.

```
{
  FILE *ifp;

  ifp = fopen(argv[1], "r"); /* open for reading  */

  copyout(ifp);              /* copy the file out */

  fclose(ifp);               /* close the file    */

  return 0;
}
```

(Question 1 continued on next page)

Student ID: ....

(Question 1 continued)

4 marks. I had:

```
/* copyout definition */

void copyout(FILE *ifp)
{
  int c;

  while((c = fgetc(ifp))!= EOF){
  putchar(c);};
}
```

There are a lot of other valid solutions. Code written in Java syntax should lose a mark!

Question 2. [12 marks]

(a) [12 marks]

You want to write a C function `lift` which takes two integer variables and sets them both to be equal to the larger. Complete the details required in the boxes below. This output should be generated:

```
$gcc test1.c
$ ./a.out
x = 1
y = 0
x = 1
y = 1
```

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

3 marks. I had:

```
/* lift prototype */
void lift(int* px, int* py);
```

The important bit is the *'s

```
int main()
{
  int x = 0;
  int y = 1;
  printf("x = %d\ny = %d\n", x, y);
```

3 marks. I had:

```
/* call to lift x and y */
  lift(&x, &y);
```

The important bit is the &'s

```
  printf("x = %d\ny = %d\n", x, y);
  return 0;
}
```

(Question 2 continued on next page)

(Question 2 continued)

6 marks: I had

```
/* lift definition */
void lift(int* i, int* j){
   if (*i < *j) *i = *j;
   else *j = *i;
}
```

Again, the important bit is the *'s

Student ID: ....

Question 3. [16 marks]

A *shape* can be either a circle, a square or a rectangle. To define the size of a circle we need to know its radius; to define a square we need to know the length of the side; to define a rectangle we need to know its length and breadth.

The following C type definitions allow us to represent a shape:

```
typedef enum {circle, square, rectangle} Geometry;

typedef struct {
    float length;
    float breadth;
} LengthBreadth;

typedef union{
    float radius;
    float side;
    LengthBreadth sides;
} Size;

typedef struct {
    Geometry shape;
    Size size;
} Shape;
```

(Question 3 continued on next page)

(Question 3 continued)

(a) [8 marks] In the box below, define a function with prototype `void printshape(Shape s);` which will print out the dimensions of a shape and whether it is a circle, rectangle or square. Here are some example outputs which `printshape` should generate:

```
Square, side 1.414214
Circle, radius 0.797885
Reactangle, 1.000000 by 2.000000
```

8 marks. I had:

```
void printshape(Shape s){
       switch(s.shape){
         case circle : printf("Circle, radius %f\n", s.size.radius); break;
         case square : printf("Square, side %f\n", s.size.side); break;
         case rectangle : printf("Reactangle, %f by %f\n",
                                 s.size.sides.length,
                                 s.size.sides.breadth); break;
     }
}
```

Important parts are correct syntax for accessing components, `printf`, and correct use of format string in `printf`. Many students will use several `if`s rather than on `switch`. In either case the control flow should be correct.
ps. I don't care if they mis-spell ″Rectangle″ as I did!

(Question 3 continued on next page)

(Question 3 continued)

(b) [8 marks] In the box below, define a function with prototype `float area(Shape s);` which will return the area of a shape:

- the area of a circle is `3.14159` times the square of the radius,
- the area of a square is the square of the length of the side,
- the area of a rectangle is the length times the breadth.

8 marks. I had (having `#defined PI` previously):

```
float area(Shape s){
        switch(s.shape){
                case circle :
                       return PI *  s.size.radius * s.size.radius;
                case square : return s.size.side * s.size.side;
                case rectangle :
                       return  s.size.sides.length * s.size.sides.breadth;
     }
}
```

Important parts are correct syntax for accessing components, and returning the correct value (Clearly I'm not bothered about whether they only have one return!). Many students will use several `if`s rather than on `switch`. In either case the control flow should be correct.

Question 4. [20 marks]

(a) [6 marks]

In the box below state what you consider to be the three most significant **advantages** which C has when compared to Java.

6 marks, hence 2 for each sensible comment. I'd expect:

- no virtual machine hence can be compiled to very efficient code
- provides programmer with the ability to access/represent data at a very low level, allowing for efficiency
- very good for e.g. distributed systems implementation due to access to OS level within programs.
- ...

You may get many others.

(Question 4 continued on next page)

(Question 4 continued)

(b) [6 marks] In the box below state what you consider to be the three most significant **disadvantages** which C has when compared to Java.

6 marks, hence 2 for each sensible comment. I'd expect:

- type system: esp. C is not OO
- lack of portability/requirement to re-compile for different architectures.
- lack of garbage collection

You may get many others. Some may even point out that many of Cs advantages are also its disadvantages!

(Question 4 continued on next page)

Student ID: ....

(Question 4 continued)

(c) [8 marks] The C compiler works in several stages:

- preprocessing,
- compilation,
- assembly, and
- linking

In the box below state what happens at the preprocessing stage, and what preprocessor directives are used for.

8 marks. I'm hoping that they will mention:

- `#include`
- `#define`
- `#ifdef` and friends

2 marks for each of those.
They should explain that pre-processing is not part of the language itself, and is basically a process of manipulation the text of the source file. They should be able to explain that `#include` loads the text of the included file, and that this helps us separate definitions from implementation and consequently helps support information hiding. They should be able to explain the use of very simple macros, and the pitfalls. They will probably struggle to give a good description `#ifdef`, and why it is used.

Question 5. [24 marks]

The following type definition and function prototypes are part of a system for recording data about students. The information about the students in a class is stored in a linked list. Privacy legislation prohibits us from keeping any more information than the students' ID numbers.

```
typedef struct student {
  int id;
  struct student *next;
} Student;
typedef Student* Class;

void printlist(Class); /* print/traverse a linked list */
int count(Class);     /* count the number of elements in the class */
void enroll(Class, int id); /* student with ID id joins the class */
void drop(Class, int id);  /* student with ID id leaves the class */
```

(a) [8 marks] Write the C code to implement the function `count` to count the number of students in the class.

8 marks. The tricky part about this data structure is that the header node is the same type as the list nodes, and all the functions. This is a bit odd, but they will not notice this, I expect.
For the count function the main points to look for are:

- that they traverse the list properly;
- that they calculate a sensible value;
- and of course they mustn't destroy the list as they traverse it (they will find that hard though)

(Question 5 continued on next page)

(Question 5 continued)

(b) [8 marks] Write C code to implement function `drop` so that the student with a given ID is removed from the class.

8 marks. The main point here is that they should free the removed node: so 3 marks for that. The other marks are for getting the algorithm and the C code correct

(Question 5 continued on next page)

(Question 5 continued)

(c) [8 marks] Write C code to implement function `enroll` so that the student with a given ID is added to from the class.

8 marks. The main point here is that they should allocate memory properly so 3 marks for that. The other marks are for getting the algorithm and the C code correct

A Operator Precedence

1 (), ->, [], .

2 ~, ++, --, + (unary), - (unary), *(unary), &(unary) , sizeof, (type)

3 *, /, % (arithmetic binary

4 +, - (arithmetic binary)

5 <<, >>

6 <, <=, >, >=

7 ==, !=

8 &

9 ^

10 |

11 &&

12 ||

13 ?:

14 =, +=, -=, ... (assignment)

15 ,

Student ID: . . . .

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

****