The Hong Kong University of Science & Technology

COMP 102: Computer Fundamentals & Programming

Spring Semester, 1996
FINAL EXAMINATION
29 May, 1996 (8:30am - 11:30am)

STUDENT NAME: ____________________________
STUDENT NUMBER: ____________________________
DEPARTMENT: ____________________________
LAB. SECTION: ____________________________

Instructions to Students

- Check that you have all 12 pages.
- Write your name, student number , department, and lab. section on this page.
- Write your student number on top right hand corner of each page.
- Answer all questions in the space provided.

For Use by Grader		
Question 1-4	__________________	/ 12
Question 5	__________________	/ 16
Question 6	__________________	/ 5
Question 7	__________________	/ 5
Question 8	__________________	/ 4
Question 9	__________________	/ 3
Question 10	__________________	/ 10
Question 11	__________________	/ 8
Question 12	__________________	/ 10
Question 13	__________________	/ 5
Question 14	__________________	/ 22
TOTAL:	__________________	/ 100

1. **(3 points)** Which of the following variable names are *valid*? Check ONLY those correct answers with a "Y"

___ i. break	___ ii. end	___ iii. not	___ iv. repeat
___ v. begin	___ vi. comp102	___ vii. 4sale	___ viii. real

2. **(3 points)** Which of the following C++ statements are valid?

```
i)  enum yesno {yes, no} yn;
ii) enum grade {'A','B','C','D','F'};
iii) enum num {a, b=3, c, d, e=1, f, g};
iv) enum num2 {a=1, b=1, c=1};
```

a. i) and ii) only
b. i) ii) and iii) only
c. i) and iii) and iv) only
d. i), ii), iii) and iv)
e. none of the above

3. **(3 points)** Suppose **i** is an integer (int i) and **A** is an array (int A[10]). The sequence of statements

```
i = 20;
A[i] = i;
```

a. causes a compile time error
b. always causes a run time error
c. has absolutely no effect on your program
d. always causes some errors in your program behaviour
e. none of the above

Circle the best of answer a to e.

4. **(3 points)** What is the output of the following program?

```
#include <iostream.h>
void main() {
     for (int i=0; i<(i+5)/2; i++) {
          cout << i;
     }
}
```

a. 0123
b. 012
c. 01
d. 0
e. none of the above

Circle the best of answer a to e.

5. **(5 points)**

a. Given the following bubble sort program

```
void bubble(int data[], int size, int& count)
{
   int temp, rightmost, current;

   count = 0;
   rightmost = size - 1;
   while (rightmost > 0) {
      current = 0;
      while ((current < rightmost)) {
         count++;
         if (data[current]>data[current+1]) {
            temp = data[current];
            data[current] = data[current+1];
            data[current+1] = temp;
         }
         current++;
      }
      rightmost--;
   }
}

#include <iostream.h>
#include <iomanip.h>
int main() {
   int A[8] = { 10,4,6,9,25,5,18,3 };
   int n;

   for (int i=0; i<8; i++)
      cout << setw(3) << A[i];

   bubble(A, 8, n);

   cout << endl;
   for (int i=0; i<8; i++)
      cout << setw(3) << A[i];

   cout << endl << n << endl;
   return 0;
}
```

Show the output:

b. **(8 points)** It is obvious that the output parameter "count" of the bubble function only depends on the size of the array "data", **not** on the content of the input array elements. That is, given an array already sorted, the program still performs the same number of comparisons. Modify the algorithm to significantly reduce the number of comparisons when the input array is already sorted. Fill in the following blanks to make your modification.

```
void bubble(int data[], int size, int& count)
{
   int temp, rightmost, current;

   int contd = __________________________________________;

   count = 0;
   rightmost = size - 1;
   while ((rightmost > 0) ____________________________________) {

      __________________________________________;
      current = 0;

      while ((current < rightmost){
         count++;
         if (data[current]>data[current+1]) {

            ___________________________________________;
            temp = data[current];
            data[current] = data[current+1];
            data[current+1] = temp;
         }
         current++;
      }
      rightmost--;
   }
}
```

c. **(3 points)** Given the above modification, if the input array is sorted and of size 10, what is the output value of "count"?

6. **(5 points)** Given the following program

```
#include <iostream.h>

main() {
     int i;
     int n = 0;
     do {
          cin >> i;
          cout << i;
          n++;
     } while ((i != -1)&&(n < 5));
     return 0;
}
```

and the input

1	2	3			4	5	6		e		-	1		4		\n								

Show the output:

STUDENT NUMBER:_____________________

7. **(5 points)** Consider the following C++ function:

```
void what(int a[], int size) {
     int tmp;
     for (int i=0; i<size/2; i++) {
          tmp = a[i];
          a[i] = a[size-i-1];
          a[size-i-1] = tmp;
     }
}
```

Explain in **no more than two** sentences (optionally, with diagram) what the above function does.

8. **(4 points)**
Fill in the blanks.
In C++, arrays are passed by ______________________________________

After passing an array, say A (int A[10]) to the following function,

```
int fct(int a[]) {
   •••
}
```

can the elements in A be altered by *fct*? (Yes or No) ____________________

9. **(3 points)** Given the *struct* declaration:

```
struct student
{
   char name[20];
   int  age;
}

student student1;
```

There is an obvious syntax error. Correct the error.

10. **(10 points)** The following function "*string2int*" is supposed to convert a string of digits into an integer. If the string contains non-digit characters before the end-of-string character, a -1 is return. Fill in the following blanks to complete the function. No predefined function should be used and no explicit ASCII code value should be included.

```
int string2int(char s[]) {

     int i = ________________________________________;

     int tmp = ______________________________________;

     int digit = ____________________________________;

     while (digit && s[i] != ________________________)

          if (_______________________________________) {

               tmp = ______________________________;
               i++;
          } else

               ____________________________________;

     if (digit)
          return tmp;
     else
          return -1;
}
```


11. **(8 points)** There is an obvious semantic bug in the following binary search program. Fix the program by modifying **no more than one line**.

```
// Searches an ordered array of integers
int binary(int data[], // input: array
           int size,   // input: array size
           int value   // input: value to find
            )           // output: index if found
                        //         otherwise return -1
{
     enum state {searching, found, absent} status;
     int lower, middle, upper, index;

     status = searching;
     lower = 0;
     upper = size;
     index = -1;

     while (status == searching) {
          middle = (lower + upper) / 2;
          if (data[middle] == value) {
               status = found;
               index = middle;
          } else if (lower > upper) {
               status = absent;
          } else if (value < data[middle]) {
               upper = middle - 1;
          } else {
               lower = middle + 1;
          }
     }
     return index;
} // binary
```

12. **(10 points)** An integer number is said to be a perfect number if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number because 6 = 1+2+3. The following program is supposed to determine and print all the perfect number between 1 and 1000. Fill in the blanks to complete the program.

```
#include <iostream.h>


____________________________________________________

main()
{
   cout << "For the integers from 1 to 1000:" << endl;

   for (int j = 2; j <= 1000; j++)
      if (perfect(j))
         cout << j << " is perfect" << endl;

   return 0;
}

int perfect(int value)
{

   int factorSum = ______________________________


   for (int i = 2; ______________________________; i++)

      if (_______________________________________)
         factorSum += i;

   if (factorSum == value)

       ____________________________________________

   else

      _____________________________________________
}
```


13. **(5 points)** Given the following program

```
#include <iostream.h>

int main() {
     char name[20];
     cin >> name;
     cout << name;
     return 0;
}
```

and the input

c	h	a	n		t	a	i		m	a	n	\n												

Show the output:

14. **(22 points)** Write a program that accepts a file name, reads text from the file and matches each word read against an input pattern. The word is outputted if it matches (partially or completely) the input pattern. You may assume that the file contains only words. Except for the first and last word in the file, words are delimited by spaces and words are made up of no more than 20 lower case letters ('a' to 'z'). You can also assume that the file name has no more than 20 characters long.

Sample run: (user input are in *italic*)

```
Input filename = word.txt
Input pattern = exa

example
exact
exa
example
texas
```

Sample file: word.txt

```
an example exact match exa an example partial match texas<eof>
```

Note: Decompose your program into appropriate modules (2 points). Clearly specify and explain the parameters in your functions (3 points). You may lose up to 10 points if your program is not easily understandable. To improve readability, please include comments and choose meaningful variable names.

IMPORTANT: You are not allowed to use any predefined string functions.

STUDENT NUMBER:_____________________

14. (Cont'd)