

TEST 2, CS 101, SECTION 3
WEDNESDAY, MARCH 24, 2004
ANSWERS

- (1) Consider the following function declaration:

```
int sum(int);
```

The function `sum` should take as input an integer `n` and return the value 0 if `n <= 0`. If `n > 0`, the value returned should be the sum $1+2+3+\dots+(n-1)+n$ of the first `n` positive integers. Write the definition of the function `sum` in the space below.

```
int sum(int n){  
  
    int i, s=0;  
  
    for (i=1; i<=n; i++)  
        s += i;  
  
    return s;  
}
```

(2) Consider the following function declaration:

```
bool leap_year (int);
```

This function should take as input an integer y and return whether year y is a non leap year. The rule for computing leap years is this:

- If y is not divisible by 4, y is not a leap year. Example: 2003 was not a leap year.
- If y is divisible by 4 and not divisible by 100, then y is a leap year. Example: 2004 is a leap year.
- If y is divisible by 100, then y is a leap year ONLY IF it is also divisible by 400. Examples: 1900 was not a leap year, but 2000 was.

Hint: Remember the mod operator `%` can be used to determine divisibility. So the logical expression `(x % 3 == 0)` tests whether an `int` variable `x` is divisible by 3.

Write the function definition in the space below.

```
bool leap_year (int y){  
  
    if (y%4)  
        return false;  
  
    /* so now we know y is divisible by 4. */  
  
    if (y%100)  
        return true;  
  
    /* so now we know y is divisible by 100. */  
  
    return (y%400 == 0);  
}
```

- (3) Consider the following program. Write the output of the program in the space given below.

```
#include <iostream.h>
int f(int);
void main() {
    cout << f(3) << endl;
}
int f(int m){
    cout << m << " ";
    if (m>0)
        return m + f(m-2);
    else
        return 0;
}
```

3 1 -1 4

- (4) Mark each of the following logical expressions as either true or false. You can assume we have declared

```
int x=2, y=2, z=4;
```

Expression	true	false
(x>=1)	X	
(z==y*x)	X	
((z<5) (y*y*y*y<z*x*x))	X	
(!(x==1) && (y==2))	X	
(y==x++)	X	

- (5) Consider the following program. What is the output? Put your answer in the space below.

```
#include<iostream.h>
void main() {
    int a=3, x=2;
    do{
        cout << a << " " << x << endl;
        if ((a==1) || (x>2)) a--;
        x += a*a;
    } while ((a!=0) && (x<20));
}
```

3 2
3 11
2 15
1 16

- (6) What is the output? Write your answer in the space below.

```
#include<iostream.h>
void jumble (int&, int&, int&, int&);
void main() {
    int a=1, b=2, c=3, d=5;
    jumble (a,b,c,d);
    cout << a << " " << b << " " << c << " " << d << endl;
}
void jumble (int& w, int& x, int& y, int& z){
    int temp = w;
    w = z; z = x; x = y; y = temp;
}
```

5 3 1 2

Multiple choice: circle the letter of the correct answer.

- (7) What is the returned value of the function `f`?

```
int f(int x, int y){return (x>y) ? y : x; }
```

- (a) The sum `x + y`.
 (b) The difference `x - y`.
 (c) The maximum of `x` and `y` (i.e. the value of the larger of `x` and `y`).
(d) The minimum of `x` and `y` (i.e. the value of the smaller of `x` and `y`).
- (8) The following code fragment should assign the `int` variable `y` to be 1 if the `int` variable `x` is 1, 3, or 5. It should assign `y` to be 0 if `x` is 2. Otherwise, it should assign `y` to be -1. Fill in the blanks with the correct keywords (your answers should be in the same order as in the code):

```
----- (x) {
    case 1: case 3: case 5:    y = 1; break;
    case 2:                    y = 0; break;
    -----:                  y = -1; break; }
```

- (a) `switch, else`.
 (b) `default, switch`.
 (c) `if, else`.
(d) `switch, default`.
- (9) The following code fragment prints out ____ asterisks.

```
for (char c='P'; c > 'N'; c--) cout << '*';
```

- (a) 0
(b) 2
 (c) 3
 (d) 4

- (10) Consider the following program. What is the output?

```
#include<iostream.h>
void main(){
    bool a=true;
    for (int b=1; b<=3; b++){
        a = !a;
        if (a) cout << '+';
    }
    if (a) cout << '-';
}
```

- (a) ++
 (b) +-
 (c) ++-
(d) +

- (11) Consider the following functions:

```
void spaces(int m){
    for (int i=1; i<=m; i++) cout << " ";
}
void line(int m){
    for (int i=1; i<=m; i++) cout << "*";
    cout << endl;
}
```

Which code fragment prints out the following pattern?

```
*****
****
***
**
*
```

- (a) for (int i=1; i<=5; i++){
 spaces(i-1); line(5-i); }
- (b) for (int i=1; i<=5; i++){
 spaces(i-2); line(6-i); }
- (c) for (int i=1; i<=5; i++){
 spaces(i-1); line(6-i); }**
- (d) for (int i=1; i<=5; i++){
 spaces(i); line(5-i); }

(12) What is wrong with the following function definition?

```
void f(int n) {  
    static int b=1;  
    b += n--;  
    if (n){  
        cout << b << endl;  
        f(n); }  
}
```

- (a) It is possible for this function to go into an infinite recursion.
- (b) There should be a semicolon after `void f(int n)`.
- (c) There is a syntax error in the line `b += n--;`.
- (d) More than 1 of (a), (b), (c) above.
- (13) Consider the following declaration, which occurs at the beginning of the file `program.cpp`, OUTSIDE OF any function definition. Which statement best describes the behavior of the variable `days`?

```
int days = 7;
```

- (a) The value of `days` can only be changed or accessed from within the `main` function.
- (b) The value of `days` can be changed or accessed from within any function defined in `program.cpp`.
- (c) The value of `days` can be accessed from within any function defined in `program.cpp`, but the value of `days` can be changed only from within the `main` function.
- (d) The value of `days` can be accessed from within any function defined in `program.cpp`, but its value can never be changed.