

1. Given the following declarations, mark each of the boolean expressions as either true (T) or false (F):

```
int a=5;  
int b=10;  
int c = 7;
```

- T **F** (a > b)
T F (b == 10)
T F (a + c > b)
T F (a < b) && (c < b)
T **F** (a == 10) || (c == 10)

2. Using the same variable declarations from 3 above, what do the following statements print.

```
if (c+b < 20)  
{  
    if (b == 5)  
    {  
        cout << "Location 1"  
    }  
    else if (b == 8)  
    {  
        cout << "Location 2"  
    }  
    else  
    {  
        cout << "location 3"  
    }  
}  
else  
{  
    cout << "location 4"  
}
```

location 3

```
for (int i=0; i<a; i++)  
{  
    cout << i*i << " "  
}
```

0 1 4 9 16

```
int x=9;  
while (x >= c)  
{  
    cout << x << " "  
    x = x - 1;  
}
```

9 8 7

3. a) Write a function which takes two integer parameters and returns the smaller of the two.

```
int min(int first, int second)    or    int min(int first, int second)
{                                  {
    if (first < second)            if (first < second)
        return first;              return first;
    else                            return second;
        return second;            }
}
```

```
or int min(int first, int second)
{
    int smaller;
    if (first < second)
        smaller = first;
    else
        smaller = second;
    return smaller;
}
```

- b) Call the function you wrote in (a) and display the larger of x and y.

```
if (min(x,y) == x)                or    int smaller = min(x, y)
    cout << y;                    if (smaller == x)
else                                cout << y;
    cout << x;                    else
                                cout << x;
```

4. Given these declarations: int a[100];
 int oddSum;
 int n;

Write a loop to calculate the sum of the odd-numbered elements (elements 1, 3, etc.) of a, storing this result in the variable oddSum.

You may assume that appropriate values have been assigned to all elements of the array.

```
oddSum = 0;                        or    oddSum = 0;
for (n=1; n<100; n+= 2)            n = 1;
    oddSum += a[n];                while (n < 100)
or                                  {
                                oddSum = oddSum + a[n];
                                n = n + 2;
                                }
```

```
oddSum = 0;
for (n=1; n<100; n++)
    if (n % 2 == 1) // a[n]% 2 would check for the array value itself being odd
        oddSum += a[n];
```

Note: for (n=1; n<100; n = n+2) is okay for the first version instead of n += 2
but for (n=1; n<100; n+2) wouldn't actually change the value of n