(1) Consider the following function declaration:

```c
int f(int n){
    if (n<=0)
        return 0;
    if (n%2)
        return 1;
    return n*n;
}
```

The function $f$ should take as input an integer $n$ and return the value 0 if $n \leq 0$. If $n > 0$ and $n$ is odd, then $f$ should return the value 1. If $n > 0$ and $n$ is even, then $f$ should return the value $n^2$. (You may assume that one of the header files `cmath` or `math.h` has been included at the beginning of the program file.)
(2) Consider the following function declaration:

```c
int num_days (int);
```

This function should take as input an integer \( m \) and return the number of days in month \( m \) in a non leap year. (For January, \( m \) is 1, and for February \( m \) is 2 and so forth.) If \( m \) is not a valid number of a month, \texttt{num_days} should return 0. So if \( m \) is 1, 3, 5, 7, 8, 10 or 12, \texttt{num_days} should return 31. If \( m \) is 2, \texttt{num_days} should return 28. If \( m \) is 4, 6, 9 or 11, \texttt{num_days} should return 30. And otherwise \texttt{num_days} should return 0.

Write the function definition in the space below:

```c
int num_days (int m){
    switch(m){
        case 1: case 3: case 5: case 7: case 8:
            return 31;
        case 2:
            return 28;
        case 4: case 6: case 9: case 11:
            return 30;
        default:
            return 0;
    }
}
```
(3) Consider the following program. Write the output of the program in the space given below.

```c++
#include <iostream.h>
int f(int);
void main() {
    cout << f(4) << endl;
}
int f(int m) {
    cout << m << " ";
    if (m)
        return m + 2*f(m-2);
    else
        return 0;
}
```

4 2 0 8

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

```c++
int x=1, y=3, z=4;
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>true</th>
<th>false</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1=x)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>(z==y*y)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>((z&lt;4)</td>
<td></td>
<td>(y+x&lt;z))</td>
</tr>
<tr>
<td>(! (x==1) &amp;&amp; (y&gt;=2))</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>(y++ == z)</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
(5) Consider the following program. What is the output? Put your answer in the space below.

```cpp
#include<iostream.h>
void main() {
    int a=3, x=0;
    do{
        cout << a << " " << x << endl;
        if (a==1) a++;
        a--;
        x += a*a;
    } while ((a!=0) && (x<7));
}
```

3 0
2 4
1 5
1 6

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

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

3 2 1
3 1 2
Multiple choice: circle the letter of the correct answer.

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

\[
\text{int } f(\text{int } x, \text{int } y)\{
\text{return } (x < y) \ ? \ x : y; \}
\]

(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) What happens if the following loop is executed as part of a program on pegasus?

\[
\text{do}\{\text{n++;} \ n=3; \ \text{cout} << \ n;\} \ \text{while} \ (\text{n}<4);
\]

(a) The computer will do nothing.
(b) The computer will hang (it will appear to do nothing and will not return to the pegasus prompt).
(c) The computer will print one 3 to the screen.
(d) The computer will print many many 3’s to the screen and not return to the pegasus prompt.

(9) The following code fragment prints out ____ asterisks.

\[
\text{for} \ \text{(char } \ c='a'; \ c < 'd'; \ c++) \ \text{cout} << '\*';
\]

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

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

\[
\text{#include<iostream.h>}
\text{void } \text{main}()\{
\text{bool } a=false;
\text{for} \ \text{(int } b=1; \ b<=3; \ b++)\{
\text{a } = \ a \ || \ (b==2);
\text{if} \ (a) \ \text{cout} << '+';
\}
\text{if} \ (a) \ \text{cout} << '-';
\}
\]

(a) ++
(b) +
(c) ++-
(d) +
(11) Consider the following functions:

```cpp
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) `line(1);
    for (int i=2; i<=4; i++){
        cout << "*"; spaces(i-2); cout << "*" << endl; }
    line(5);`

(b) `for (int i=1; i<=4; i++){
        cout << "*"; spaces(i-2); cout << "*"; }
    line(5);`

(c) `for (int i=1; i<=5; i++) line(i);`

(d) `line(1);
    line(2);
    for (int i=3; i<=5; i++){
        cout << "*"; spaces(i-2); cout << "*"; }
```
(12) What is wrong with the following function definition?

```c
void f(int n)
{
    while (n!=0) {
        n--;
        cout << n << endl;
    }
}
```

(a) Nothing.

(b) It is possible for this function to go into an infinite loop.

(c) There should be a semicolon after `void f(int n)`.

(d) Both (b) and (c).

(13) What is the output of the following program?

```c
#include<iostream.h>
int f(int);
void main(){
    cout << f(1) << endl << f(2) << endl;
}
int f(int a){
    static int b=1;
    b++;
    return a+b;
}
```

(a) 2

(b) 3

(c) 3

(d) None of the above.