1. **(4 marks)** Provide the results of the following expressions:

   \[
   \begin{array}{cccccccccc}
   x & y & x \&\& y & y \&\ (y-1) & x \mid\mid y & x \mid y & x \wedge y & x \ll 1 & (x>y) ? x: y & '0'+x \\
   1 & 3 & 1 & 2 & 1 & 3 & 2 & 2 & 3 & 49 \\
   \end{array}
   \]

2. **(8 marks)** We can implement a block of C code as a macro, an inline function or a regular function. Circle the one(s) that has/have the given properties.

   - The code is handled by C pre-processor: macro; inline function; regular function.
   - The code has the worst run-time efficiency: macro; inline function; regular function.
   - The errors in the code are checked: macro; inline function; regular function.
   - The code has to be defined in the header, not the source file: macro; inline function; regular function.
   - The code will increase the size of the executable: macro; inline function; regular function.

3. **(3 marks)** The following C function locates the position of a character in a word. If the character occurs in the word, it returns the first position (0 and up) where the character occurs. Otherwise, it returns -1. Identify the error, if any, in the code.

   ```c
   int locate(char word[], char c)
   {
      int pos;
      int found=0;
      for (pos=0; pos < strlen(word) && !found; pos++)
         if(word[pos]==c)
            found=1;
      if (found) return pos; else return -1;
   }
   
   Answer: return pos should be return pos-1
   ```
4. (4 marks) Give the outputs of the following code fragments.

<table>
<thead>
<tr>
<th>C code</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>int i=0;</td>
<td>6</td>
</tr>
<tr>
<td>while (i++ &lt; 5);</td>
<td></td>
</tr>
<tr>
<td>printf(&quot;%d \n&quot;,i);</td>
<td></td>
</tr>
<tr>
<td>double a = 2; int n = 5;</td>
<td></td>
</tr>
<tr>
<td>double r = 1; double b = a;</td>
<td></td>
</tr>
<tr>
<td>int i = n;</td>
<td></td>
</tr>
<tr>
<td>while (i &gt; 0)</td>
<td></td>
</tr>
<tr>
<td>if(i %2 == 0)</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>b = b * b;</td>
<td></td>
</tr>
<tr>
<td>printf(&quot;b is %d \n&quot;,b);</td>
<td></td>
</tr>
<tr>
<td>i = i / 2;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
<tr>
<td>else</td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>r = r * b;</td>
<td></td>
</tr>
<tr>
<td>printf(&quot;r is %d \n&quot;,r);</td>
<td></td>
</tr>
<tr>
<td>i = i - 1;</td>
<td></td>
</tr>
<tr>
<td>};</td>
<td></td>
</tr>
</tbody>
</table>

5. (2 marks) Which one of the following statements is not equivalent to the other two, assuming the loop bodies are the same?

(a) while (i < 10) {...}  
(b) for ( ; i < 10; ) {...}  
(c) do {...} while (i < 10);  

Answer: (c)  

6. (10 marks) Implement C code to create a 2-dimensional integer array and then use loops to assign the following values:

<table>
<thead>
<tr>
<th>13</th>
<th>9</th>
<th>5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>10</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>12</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

Implement C code to print the array contents as followings:  
4, 8, 12, 16, 3, 7, 11, 15, 2, 6, 10, 14, 1, 5, 9, 13  
You can manipulate your array using indices, constant pointers or pointer variables.
int table[4][4];
int i, j;
int value=1;

for (j=3; j>=0; j--)
for (i=0; i< 4; i++)
    table[i][j]=value++;

for (i=3; i>=0; i--)
for (j=3; j>=0; j--)
    printf(‘%d, ’, table[i][j]);

7. (6 marks) Give the outputs of the following C code:

```
#include<stdio.h>
struct point {int x; int y};
void swap_struct(struct point s)
{
    int temp; temp=s.x; s.x=s.y; s.y=temp;
}
void swap_array(int a[2])
{
    int temp=a[0]; a[0]=a[1]; a[1]=temp;
}
main()
{
    struct point first={10,20};
    struct point second=first;
    int arrOfTwo[2]={100,200};
    swap_struct(first);
    swap_array(arrOfTwo);
    printf(‘first: %d, %d 
’, first.x, first.y);
    printf(‘second: %d, %d 
’, second.x, second.y);
    printf(‘arrOfTwo: %d, %d 
’, arrOfTwo[0], arrOfTwo[1]);
}
```

Answer:

first: 10, 20
second: 10, 20
arrOfTwo: 200, 100

8. (3 marks) What output does the following C code produce?

```
int i=3
switch (i % 3) {
    case 0: printf(‘zero 
’);
    case 1: printf(‘one 
’);
    case 2: printf(‘two 
’);
}
```
9. (10 marks) Given the following C code:

```c
int a[] = {5, 15, 34, 54, 14, 2, 52, 72};
int *p = &a[1], *q = &a[5];
```

- What is the value of `(p + 2)`? 54
- What is the value of `p + 2`? 17
- What is the value of `a + 2`? 7
- What is the value of `(a + 2)`? 34
- What is the value of `(q - 2)`? 54
- What is the value of `p - q`? -4
- Is the condition `q > p` true or false? True
- Is the condition `p > q` true or false? True
- Is the condition `p + 2 == q - 2` true or false? True
- Is the condition `(p + 2) == (q - 2)` true or false? True

Answer:
- zero
- one
- two