<table>
<thead>
<tr>
<th>Question #</th>
<th>Total Possible</th>
<th>Total Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
For each of the following exam questions, place your answers on the question paper. There will be no scrap paper given. If you need scrap, two blank pages are provided at the back of this exam paper. Your name must appear at the top of each page, including the scrap pages in the back.

CALCULATORS and CELL PHONES MAY NOT BE USED DURING THE EXAM!!

1. (10 points) Match the term in **COLUMN A** with the phrase in **COLUMN B** which defines the term. Write the letter of the phrase in the blank space to the left of the term. Note that as there are more phrases than terms, not every phrase matches a term.

<table>
<thead>
<tr>
<th>COLUMN A</th>
<th>COLUMN B</th>
</tr>
</thead>
<tbody>
<tr>
<td>___1. ASCII</td>
<td>A. The set of rules for formulating grammatically correct language statements.</td>
</tr>
<tr>
<td>___2. compiler</td>
<td>B. Contains the fundamental instructions that initially control the computer. These instructions cannot be lost or changed by the computer user.</td>
</tr>
<tr>
<td>___3. input device</td>
<td>C. a data type with two values: true and false.</td>
</tr>
<tr>
<td>___4. CPU</td>
<td>D. an encoding of characters into a sequence of 0s and 1s</td>
</tr>
<tr>
<td>___5. syntax</td>
<td>E. The machine language version of the high level language program.</td>
</tr>
<tr>
<td>___6. header file</td>
<td>F. Volatile memory where your programs and data are stored while you use the computer.</td>
</tr>
<tr>
<td>___7. algorithm</td>
<td>G. libraries containing functions and symbols that can be included in a C++ program using preprocessor directives</td>
</tr>
<tr>
<td>___8. bool</td>
<td>H. A program which translates instructions written in a high-level language into the equivalent machine language</td>
</tr>
<tr>
<td>___9. semicolon</td>
<td>I. examples are keyboard, mouse, USB drive</td>
</tr>
<tr>
<td>___10. bug</td>
<td>J. Used as a terminator of each statement in C++ source code</td>
</tr>
<tr>
<td></td>
<td>K. Hardware which performs control functions and arithmetic and logical operations</td>
</tr>
<tr>
<td></td>
<td>L. An error in a program</td>
</tr>
<tr>
<td></td>
<td>M. A step by step problem solving process in which a solution is arrived in a finite amount of time.</td>
</tr>
</tbody>
</table>
2. (15 points) What output will be displayed from each of the following program segments?

a) (4 pts)
```cpp
int x = 17;
while (x > 3)
{
    x = x - 5;
    cout << x << endl;
}
cout << '$' << x << '$' << endl;
```

b) (4 pts)
```cpp
const int NUM = 4;

int i, j;

for (i = 1; i <= NUM; i++)
{
    for (j = 1; j <= 2; j++)
    {
        cout << setw(3) << i * j;
    }
    cout << endl;
}
```
c) (3 pts)
int x;
double y;

x = (11 % 3) * 2 - 3 / 2 + 1;
y = (5 / 2.0 + 1.5 + 3) * 2 / 4 + 1;
cout << x << 't' << y << endl;

d) (4 pts)
int list[7] = {1, 2, 0, -6, 7, 9, 12};
int i = 2;

while (i <= 3) {
    list[i] = list[i] + i;
    i++;
}

for (i = 0; i < 7; i++)
    cout << list[i] << " ";
3. (15 points) Willowbrook College has a terrific baseball team of 7 players. The following array represents the total number of homeruns scored by each player:

```c
int homeruns[7] = { 58, 150, 0, 234, 6, 98, 43 };  
```

a. (3 pts) Draw the array in a diagram.

b. (6 pts) Write the code to compute the total number of homeruns scored by the team this year and to determine the maximum number produced by the leading homerun player.

c. (6 pts) Write the code to compute the average number of homeruns scored by a player. Print the average.
4. (10 points)
   a. (2 points) Write a declaration for the array below and name it matrix.

   
   \[
   \begin{array}{ccc}
   -1 & 7 & 1 \\
   3 & 4 & -1 \\
   1 & 3 & 5 \\
   6 & -3 & 1 \\
   \end{array}
   \]

   b. (4 points) Given the contents of matrix above, what is printed by this code?

   ```c
   int i, j, sum;
   for ( j = 0; j < 3; j++)
   {
      sum = 0;
      for (i = 0; i < 4; i++)
         sum = sum + matrix[i][j];
      cout << "sum #" << j + 1 << " = " << sum << endl;
   }
   ```

   c. (4 points) Suppose that the contents of matrix changed. Write the code that will print out all elements of matrix that are negative.

   ```c
   ```
5. (10 points) Sub-questions a – d are 1 point each.

Suppose that you have the following declarations:

```c
char lineA[20] = "END OF";
char lineB[20] = "SPR 2011";
char lineC[20];
int x1, x2;
```


b. `x1 = strlen( lineB );` What value does `x1` receive? ______________

c. `x2 = strcmp( lineB, "YEAR" );`
   What is the value of `x2`? ______________

d. After `strcpy( lineC, lineA )` what is in `lineC`? ______________

e. (6 points) Suppose there was a `lineD` declared similar to `lineA` or `lineB` above.

   Further assume that the initial value stored is not known to you. Write a code fragment that finds the number of occurrences of the letter 'E' in `lineD`. DO NOT WRITE AN ENTIRE PROGRAM.

   EXAMPLE: If `lineD` stored the phrase “HELLO THERE”, your code would display the following output:

   There are 3 E’s in lineD.
6. (10 points) A value-returning function, `weeklyWage` computes and returns the total money that a part-timer makes for a week. The function accepts two parameters: an integer parameter (total hours), a double parameter (pay rate per hour).

Write the function **prototype** for `weeklyWage`

```

```

Write an example **call to the function**.

```

```

Write the **function definition (Header and Body)** for `weeklyWage`

```

```
7. (10 points) What is printed by the following program?

```cpp
#include <iostream>
using namespace std;

bool fun1( int x, int y );
void fun2( int x, int& y );
int main( )
{
    int s = 3, t = 7;
    bool retVal;
    cout << "Before fun1: s = " << s << " t = " << t << endl;
    retVal = fun1( s, t );
    if( retVal == true )
    
    cout << "Return true\n";
    cout << "After fun1: s = " << s << " t = " << t << endl << endl;
    s = 9, t = 5;
    cout << "Before fun2: s = " << s << " t = " << t << endl;
    fun2( s, t);
    cout << "After fun2: s = " << s << " t = " << t << endl;
    return 0;
}

bool fun1( int x, int y ) {  
    x = x + 3;
    y = y - 3;
    if( x > y )
    return true;
    return false;
}

void fun2( int x, int& y ) {  
    int tmp = x;
    x = y;
    y = tmp;
} 
```
8. (20 points) Write a complete program to determine some statistics of email messages data

A data file named eMail.txt exists and contains in it the type of email (either character C or W for College or Work), the sending account, and the size in KB (kilo bytes) of the account. An example of a few lines of data is as follows:

<table>
<thead>
<tr>
<th>Type</th>
<th>Sender</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td><a href="mailto:john@nyu.edu">john@nyu.edu</a></td>
<td>3</td>
</tr>
<tr>
<td>W</td>
<td><a href="mailto:sals@supermarket.com">sals@supermarket.com</a></td>
<td>13</td>
</tr>
<tr>
<td>W</td>
<td><a href="mailto:jackie@ymail.com">jackie@ymail.com</a></td>
<td>45</td>
</tr>
<tr>
<td>C</td>
<td><a href="mailto:sue@csi.cuny.edu">sue@csi.cuny.edu</a></td>
<td>4</td>
</tr>
</tbody>
</table>

Assume we don’t know the exact amount of lines of data in the file in advance; write a complete C++ program (put your code on the next page) that will do the following.

i) Open the file and read in the data.

ii) Write a value returning function that classifies the email by the size as “LARGE” or “MEDIUM” or “SMALL” by returning the character L, M, or S. Note: less than 5 KB is SMALL, between 5 and 20 KB is MEDIUM, and over 20 KB is LARGE.

iii) Determine the following totals:
   - total number of email messages,
   - total space that all the messages take up in KB,
   - total number of SMALL messages.

iv) Print the output so that it is organized as below. (Of course there will be more lines, because this is shown only for the lines above). All printing is done in the main function.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>SENDER</th>
<th>SIZE</th>
<th>CLASSIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td><a href="mailto:john@nyu.edu">john@nyu.edu</a></td>
<td>3</td>
<td>SMALL</td>
</tr>
<tr>
<td>W</td>
<td><a href="mailto:sals@supermarket.com">sals@supermarket.com</a></td>
<td>13</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>W</td>
<td><a href="mailto:jackie@ymail.com">jackie@ymail.com</a></td>
<td>45</td>
<td>LARGE</td>
</tr>
<tr>
<td>C</td>
<td><a href="mailto:sue@csi.cuny.edu">sue@csi.cuny.edu</a></td>
<td>4</td>
<td>SMALL</td>
</tr>
</tbody>
</table>

Total emails: 4
Total KBs: 65
Total SMALL emails: 2
Program code for Question 8 goes here: