READ THIS NOW!

Failure to read and follow the instructions below may result in severe penalties. Failure to adhere to these directions will not constitute an excuse or defense.

- Print your name in the space provided below.
- Print your name and ID number on the Opscan form; be sure to code your ID number on the Opscan form.
  Code Form A on the Opscan; code your room group number: Nor 136, 8:00 am = 1, Nor 136, 11:00 am = 2, Squires Colonial, 8:00 am = 3.
- Choose the single best answer for each question — some answers may be partially correct. If you mark more than one answer, it will be counted wrong.
- Unless a question involves determining whether given C++ code is syntactically correct, assume that it is. The given code has been compiled and tested, except where there are deliberate errors. Unless a question specifically deals with compiler #include directives, you should assume the necessary header files have been included.
- Note that questions about printed values disregard formatting completely.
- Be careful to distinguish integer values from floating point (real) values (containing a decimal point). In questions/answers which require a distinction between integer and real values, integers will be represented without a decimal point, whereas real values will have a decimal point, [1044 (integer), 1044.0 (real)].
- When you have finished, sign the pledge at the bottom of this page and turn in the test and your Opscan.
- This is a closed-book, closed-notes examination. No calculators or other electronic devices may be used during this examination. You may not discuss (in any form: written, verbal or electronic) the content of this examination with any student who has not taken it. You must return this test form when you complete the examination. Failure to adhere to any of these restrictions is an Honor Code violation.
- There are 40, 2.5-point multiple-choice questions.
- The answers you mark on the Opscan form will be considered your official answers.

Do not start the test until instructed to do so!

Print Name (Last, First) _______________________________________________________

Pledge: On my honor, I have neither given nor received unauthorized aid on this examination.

__________________________________________  ________________________________

signature
For the next two questions, consider the execution of the following code fragment:

```cpp
int I = 6, J = 0;
while (I >= 4){
    I--;
    J = J - I;
}
cout << "I = " << I << endl;
cout << "J = " << J << endl;
```

1) What value is printed for the variable I:
   1) 1  2) 2  3) 3  4) 4  5) 5  6) 6  7) 0  8) none of the above

2) What value is printed for the variable J:
   1) 0  2) -5  3) -9  4) -12  5) -14  6) -18  7) 6  8) none of the above

Consider executing the following code fragment (assume any additional declarations, etc, needed to make the code syntactically correct):

```cpp
int j = 7;
double varq = 1.0, newq = 0.0;
while (j < 75) {
    newq = j + varq;
    j = j + 6;
}
```

3) How many times will the body of the loop be executed?
   1) 7  2) 11  3) 12  4) 13  5) 75  6) none of the above

4) What is the value of j after the last iteration of the loop?
   1) 7  2) 14  3) 73  4) 75  5) 79  6) 85  7) none of the above
For the next three questions, assume the input file stream \texttt{ifile} is connected to a file containing the following data:

\begin{verbatim}
4  1  7  5  3  -2
\end{verbatim}

Consider the execution of the code fragment given in each question and determine the value that would be printed. Choose from the following answers:

5) \begin{verbatim}
int sum, count, mystery, value, total = 0;
ifile >> mystery;
sum = mystery;
for (count = 0; count <= total; count++) {
  ifile >> value;
  sum = sum + value;
}
cout << "sum=" << sum;
\end{verbatim}

6) \begin{verbatim}
int sum = 0, mystery, count = 1, value;
ifile >> mystery;
value = mystery;
do {
  ifile >> mystery;
  sum = sum + mystery;
  count++;
} while (count < value);
cout << "sum=" << sum;
\end{verbatim}

7) \begin{verbatim}
int sum = 0, mystery, value = 0;
ifile >> mystery;
while ((value > -1) && (mystery > 0)) {
  --mystery;
  sum = sum + value;
  ifile >> value;
}
cout << "sum=" << sum;
\end{verbatim}

Consider executing the following code fragment (assume any additional declarations, etc, needed to make the code syntactically correct):

\begin{verbatim}
int j = 28;
while (j != 13) {
  cout << j << endl;
  j = j - 3;
}
\end{verbatim}

8) How many times will the body of the loop be executed?

1) 4  2) 5  3) 6  4) infinite loop  5) none of the above

9) What is the last value printed?

1) 28  2) 25  3) 22  4) 19  5) 16  6) 13  7) 10  8) none of these
Consider executing the following program:

```cpp
#include <iostream.h>
void main() {
    int i, j, sum;
    for (i = 0; i < 3; i++) {
        sum = 0;
        for (j = 0; j < 2; j++)
            sum = sum + i + j;
        cout << sum << endl;
    }
}
```

10) What is the value printed on the third line of output?

1) 0  
2) 1  
3) 2  
4) 3  
5) 4  
6) 5  
7) 6  
8) none of the above

Now, consider executing the following slightly different program:

```cpp
#include <iostream.h>
void main() {
    int i, j, sum;
    sum = 0;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 2; j++)
            sum = sum + i + j;
        cout << sum << endl;
    }
}
```

11) What is the value printed on the last line of output?

1) 0  
2) 1  
3) 2  
4) 3  
5) 4  
6) 5  
7) 6  
8) none of the above

Assume the following declarations:

```cpp
void Fix(double & realvar, int intvar);
int someInt = 42;
double someFloat = 3.14;
```

12) Which of the following would represent an appropriate call(s) of the function `Fix`?

1) Fix(6.85, 24);  
2) Fix(6.85, someInt);  
3) Fix(someFloat, 24);  
4) Fix(someFloat, someInt);  
5) Fix(someFloat, someInt + 5);  
6) all of the above  
7) 1 and 2 only  
8) 4 and 5 only  
9) 3, 4 and 5 only  
10) none of the above
13) A function, `someFunc`, has two formal parameters, `A1` and `B2`, both of type `int`. The data flow (communication) for variable `A1` is two-way, into and out of the function. The data flow for variable `B2` is one-way, into the function. Which of the following is the most appropriate prototype for `someFunc`?

1) `void someFunc(int A1, int B2);`
2) `void someFunc(int& A1, int B2);`
3) `void someFunc(int A1, int& B2);`
4) `void someFunc(int& A1, int& B2);`
5) all of the above
6) none of the above

For the next three questions, consider execution of the following program:

```cpp
void DoThis(int Alpha, int& Beta);

void main()
{
    int Tmp = 15;
    int Ben = -5, Jer = 42;
    DoThis(Ben, Jer);
    cout << “Ben = ” << Ben << endl;
    cout << “Jer = ” << Jer << endl;
    cout << “DoThis Tmp = ” << Tmp << endl;
}

void DoThis(int Alpha, int& Beta) {
    int Tmp;
    Alpha = Alpha + 100;
    Tmp = Beta;
    Beta = 999;
}
```

14) What value is printed for the variable `Ben`?

1) -5  2) 15  3) 42  4) 95  5) 142
6) 999  8) None of the above

15) What value is printed for the variable `Jer`?

1) -5  2) 15  3) 42  4) 95  5) 142
6) 999  8) None of the above

16) What value is printed for the variable `Tmp`?

1) -5  2) 15  3) 42  4) 95  5) 142
6) 999  8) None of the above

17) Given the declaration

```cpp
char table[7][9];
```

which of the following stores the character ‘B’ into the sixth column and third row of the array?

1) `table[3][6] = ‘B’;`
2) `table[6][3] = ‘B’;`
3) `table[2][5] = ‘B’;`
4) `table[5][2] = ‘B’;`
5) `table[4][7] = ‘B’;`
6) `table[7][4] = ‘B’;`
7) `table[3] = ‘B’;`
8) None of these
For the next four questions, consider the incomplete function definition given below:

```
// SmallValue() takes an array of integers and the number of values
// it contains, and returns the value of the smallest item in the array,
// Parameters:
// List[] array of integers
// HowMany number of values stored in List[]
// Returns:
// the smallest value in the array
//
int SmallValue (________ List[], ______ HowMany) { // line A
    int smallest; // line B
    int Lo = 0;    // line C
    smallest = ______; // line D
    for (Lo=1; Lo < HowMany; Lo++) { // line E
        if (List[Lo] < smallest) { // line F
            smallest = List[______]; // line G
        }
    }
    return smallest; // line H
}
```

18) How should the blank preceding the first parameter in line A be filled?

1) const int  
2) int  
3) int&  
4) it should be left blank  
5) none of these

19) How should the blank preceding the second parameter in line A be filled?

1) const  
2) int  
3) int&  
4) it should be left blank  
5) none of these

20) How should the blank in line D be filled?

1) Lo  
2) List[Lo]  
3) Lo - 1  
4) List[Lo-1]  
5) none of these

21) How should the blank in line G be filled?

1) smallest  
2) List[Lo]  
3) HowMany  
4) List[HowMany]  
5) none of these

(Hint – try executing the function for a small array.)
22) What is the output of the following program?

```cpp
#include <iostream.h>
void main() {
    int Ray[5] = {100,200,300,400,500};
    int k;
    for (k = 1; k < 5; k++)
        cout << Ray[k-1] << ' ';
}
```

1) 100 200 300 400 500
2) 100 200 300 400 0
3) 500 400 300 200 100
4) 2 3 4 5
5) 1 2 3 4 5
6) 400 300 200 100
7) 100 200 300 400
8) none of the above

For the next four questions, consider writing a program that contains the following variable declarations and function prototypes.

```cpp
const int Size = 5;
void InitArray (int Ray[], int Dim);
int aRay[Size] = {1, 2, 3, 4, 5},
bRay[Size] = {6, 7, 8, 9, 10},
cRay[Size-1] = {0};
```

Indicate whether the proposed statement given in each question is:

1) syntactically illegal; i.e., there would be a compile-time error
2) syntactically legal, but logically incorrect
3) syntactically legal and logically correct, as far as we can tell

23) aRay[Size-1] = bRay[Size-1];
24) InitArray(cRay, Size);
25) InitArray(Size, cRay[]);
26) cRay[aRay[1]] = bRay[Size-aRay[0]];

For the next 4 questions, assume the following array declarations are in effect. Treat each question independently of the others.

```c
char Line1[35] = "2 B || "; // no space between vertical bars
char Line2[35] = "! 2B, i.e. the ?";
```

27) Suppose the following statement is executed:

```c
strcat(Line1, Line2);
```

What is now stored in the array Line1, (omitting the quotes)?

1) "! 2B, i.e. the ?"
2) "2 B || ">
3) "! 2B, i.e. the ?2B || "
4) "2 B || ! 2B, i.e. the ?"
5) none of the above

28) Suppose the following statement is executed:

```c
int HowBig = strlen(Line1);
```

What is the value of the variable HowBig?

1) 6
2) 7
3) 8
4) 0
5) none of the above

29) Suppose the following statement is executed:

```c
strcpy(Line1, Line2);
```

What is now stored in the array Line1, (omitting the quotes)?

1) "! 2B, i.e. the ?"
2) "2 B || ">
3) "! 2B, i.e. the ?2B || "
4) "2 B || ! 2B, i.e. the ?"
5) none of the above

30) Suppose the following statement is executed:

```c
int order = strcmp("Daniel", "Danny");
```

What is the value of the variable order?

1) < 0 (a negative int value)
2) > 0 (a positive int value)
3) 0 (zero)
4) none of the above