For questions 1 through 5, consider the following implementation of a function to find the maximum value in an array of integers. The for loop is required to be implemented using pointers to access elements rather than direct array indexing.

```c
int maxEntry(const int* const Data, int Sz) {                // Line 1
    if ( Data == NULL || Sz <= 0 ) return INT_MIN;            // 2
    int Count = 0;                                            // 3
    // Set hiSoFar to point to the first array element:
    const int *hiSoFar = __________;                           // 4
    // Set Current to point to the second array element:
    const int *Current = __________;                           // 5
    for ( ; Count < Sz; __________ ) {                        // 6
        if ( __________ )                                  // 7
            hiSoFar = Current;                              // 8
    }
    return ( __________ );                                   // 9
}
```

1. How should the blank in Line 4 be filled?
   1) Data  
   2) *Data  
   3) &Data  
   4) &Data[0]  
   5) Data[0]  
   6) 3 or 5 only  
   7) 3 or 4 only  
   8) None of these

2. How should the blank in Line 5 be filled?
   1) hiSoFar  
   2) hiSoFar++  
   3) Data++  
   4) &Data[1]  
   5) Data[1]  
   6) 2 or 4 only  
   7) 2 or 5 only  
   8) None of these

3. How should the blank in Line 6 be filled?
   1) Count++  
   2) Current++  
   3) Count++, Current++  
   4) It should be left blank.  
   5) None of these
4. How should the blank in Line 7 be filled?

1) \text{Current} > \text{hiSoFar} \quad 4) \text{*Current} < \text{*hiSoFar}
2) \&\text{Current} > \&\text{hiSoFar} \quad 5) \text{None of these}
3) \text{*Current} > \text{*hiSoFar}

5. How should the blank in Line 9 be filled?

1) \text{*hiSoFar} \quad 4) \text{It should be left blank.}
2) \&\text{hiSoFar} \quad 5) \text{None of these}
3) \text{hiSoFar}

For questions 6 through 8, assume that $P$ and $Q$ are pointers of the same type, and that each has been assigned a value.

6. What comparison would determine whether $P$ and $Q$ have targets with the same value?

1) \&P == \&Q \quad 4) \text{All of them}
2) \*P == \*Q \quad 5) 1 and 2 only
3) $P == Q$ \quad 6) 1 and 3 only

7. What comparison would determine whether $P$ and $Q$ have the same target?

1) \&P == \&Q \quad 4) \text{All of them}
2) \*P == \*Q \quad 5) 1 and 2 only
3) $P == Q$ \quad 6) 1 and 3 only

8. What comparison would determine whether $P$ and $Q$ store the same value?

1) \&P == \&Q \quad 4) \text{All of them}
2) \*P == \*Q \quad 5) 1 and 2 only
3) $P == Q$ \quad 6) 1 and 3 only

9. Assume the variable declarations:

```c
int Foo = 0;
int *ptr = &Foo;
```

Which of the following statements will change the value of $\text{Foo}$ to 1?

1) $\text{ptr++}$; \quad 5) \text{All of these}
2) $\text{Foo++}$; \quad 6) 1 and 2 only
3) $(\text{*Foo})++$; \quad 7) 1 and 4 only
4) $(\text{*ptr})++$; \quad 8) 2 and 4 only

9) 3 and 4 only
10) \text{None of these}
For questions 10 through 13 assume that we have a dynamically allocated array \( A \) of integers of dimension \( \text{Size} \), with memory layout as shown:

\[
\begin{array}{|c|c|}
\hline
\text{Index} & \text{Address} \\
\hline
0 & 007D0E70 \\
1 & 007D0E74 \\
2 & 007D0E78 \\
3 & 007D0E7C \\
4 & 007D0E80 \\
\hline
\end{array}
\]

10. Which code fragment(s) could be inserted in the blank in order to safely initialize each element of \( A \) to zero?

\[
\text{int* p = &A[0];} \\
\text{for (int Idx = 0; Idx < Size; Idx++, p++) { } }
\]

1) \( *A = 0; \)
2) \( A[Idx] = 0; \)
3) \( *p = 0; \)
4) \( *Idx = 0; \)
5) All of the above
6) 1 and 2 only
7) 1 and 3 only
8) 1 and 4 only
9) None of these

11. What value will be printed by the code fragment:

\[
\text{for (int Idx = 0; Idx < Size; Idx++) { } } \\
\text{A[Idx] = int(&A[Idx]);} // typecast converts address to int \\
\text{cout << hex << A[3] << endl; // manipulator causes hex output}
\]

1) 007D0E70
2) 007D0E74
3) 007D0E78
4) 007D0E7C
5) 007D0E80
6) Unknown
7) None of these

12. Assuming only the initial declarations given above, what logical error(s) would result if the following statement were executed: \( A = \text{new int}[2 \times \text{Size}]; \)

1) A dangling pointer would result (a pointer whose value is the address of memory that the program no longer owns).
2) A memory leak would result (the program would own memory that it could no longer access).
3) Both a dangling pointer and a memory leak would result.
4) Neither a dangling pointer nor a memory leak, but some other logical error would result.
5) No logical error would result.

13. Assuming only the initial declarations given above, and execution of the correctly completed code given in question 11, what logical error(s) would result if the following statement were executed: \( \text{delete } [\ ] p; \)

1) A dangling pointer would result (a pointer whose value is the address of memory that the program no longer owns).
2) A memory leak would result (the program would own memory that it could no longer access).
3) Both a dangling pointer and a memory leak would result.
4) Neither a dangling pointer nor a memory leak, but some other logical error would result.
5) No logical error would result.
14. Consider implementing a function to dynamically allocate an array of integers and set all its elements to zero:

```cpp
void ZeroIt(int* A, const int Size) {
A = new int[Size];
for (int Idx = 0; Idx < Size; Idx++) {
    A[Idx] = 0;
}
}
```

Which of the following choices for the blank preceding the formal parameter `A` is best?

1) `int*`  
2) `int*&`  
3) `const int*`  
4) `int* const`  
5) `const int* const`  
6) All of the above

15. Which of the following statements about C++ classes is false?

1) Classes can have private member functions.  
2) Classes can have public, private and protected members.  
3) By default, members of classes are public.  
4) Classes can have public data members.  
5) Aggregate assignment is permitted for classes.  
6) None of these, (all are true).

16. Which of the following C++ built-in operations are automatically defined for class objects?

1) `==`  
2) `=`  
3) `<<`  
4) 1 and 2 only  
5) 1 and 3 only  
6) 2 and 3 only  
7) All of them  
8) None of these

For questions 17 through 21, consider the class declaration:

```cpp
class Farey {
private:
    int Top,
    Bottom;

public:
    Farey();
    Farey(int T, int B);
    Farey operator+(const Farey& RHS) const;
    Farey operator-(const Farey& RHS) const;
    bool operator==(const Farey& RHS) const;
    void Display(ostream& Out) const;
};
```

Farey::Farey() {
    Top = Bottom = 0;
}

Farey::Farey(int T, int B) {
    Top = T;
    Bottom = B;
}
Farey Farey::operator+ (const Farey& RHS) const {
    return Farey(Top + RHS.Top, Bottom + RHS.Bottom);
}

Farey Farey::operator- (const Farey& RHS) const {
    return Farey(Top - RHS.Top, Bottom - RHS.Bottom);
}

bool Farey::operator==(const Farey& RHS) const {
    return (Top == RHS.Top) && (Bottom == RHS.Bottom);
}

void Farey::Display(ostream& Out) const {
    Out << Top << '/' << Bottom;
}

Again, assuming everything necessary is in scope, consider the following code fragment:

```cpp
Farey A(3, 5), B(1, 4), C(2, 4), D(0, 5), E;
E = A + B;                      // line 1
A.Display(cout);                //      2
E = A + B - C;                  //      3
E = 2*A;                        //      4
```

17. After the execution of line 1, what are the values of E.Top and E.Bottom, respectively?
   1) 0 and 0  
   2) 3 and 5  
   3) 1 and 4  
   4) 4 and 9  
   5) Unknown  
   6) None of these

18. What is written to the stream cout when line 2 is executed?
   1) "3/5"  
   2) Nothing  
   3) None of these

19. After the execution of line 3, what are the values of E.Top and E.Bottom, respectively?
   1) 0 and 0  
   2) 4 and 9  
   3) 2 and 5  
   4) The statement isn't allowed.  
   5) Unknown  
   6) None of these

20. After the execution of line 4, what are the values of E.Top and E.Bottom, respectively?
   1) 6 and 10  
   2) 6 and 5  
   3) 3 and 10  
   4) The statement isn't allowed.  
   5) Unknown  
   6) None of these
Consider the following code fragment:

```cpp
Farey X(1, 2), Y(2, 4);
if ( X + X == Y )                  // line 5
    cout << "X + X == Y" << endl;
else
    cout << "X + X != Y" << endl;
```

21. When the if statement beginning in line 5 is executed, what is written to `cout`?

1) "X + X == Y"  
2) "X + X != Y"

For questions 22 through 25, consider the following class and partial implementation:

```cpp
class Quadratic {
private:
    double Coefficient[3];
public:
    Quadratic(double a = 0.0, double b = 0.0, double c = 0.0);
    double Evaluate(double x) const;
    Quadratic operator+(const Quadratic& RHS) const;
    void Display(ostream& Out) const;
};
Quadratic::Quadratic(double a, double b, double c) {
    Coefficient[0] = a;
    Coefficient[1] = b;
    Coefficient[2] = c;
}
double Quadratic::Evaluate(double x) const {
    return ( Coefficient[0]*x*x + Coefficient[1]*x + Coefficient[2] );
}
```

22. Given the declaration: `Quadratic F(1, 2, 3);`

What value is output by the statement: `cout << F.Evaluate(2);`

1) 6  
2) 11  
3) 17  
4) None of these

23. Given the declaration: `Quadratic G(1);`

What value is output by the statement: `cout << G.Evaluate(2);`

1) 1  
2) 4  
3) Not allowed.  
4) None of these
A designer wants to add an addition operation to the class Quadratic. Consider the partial implementation:

```cpp
Quadratic::operator+(const Quadratic& RHS) const { // line 1
    double a = Coefficient[0] + RHS.Coefficient[0];
    double b = Coefficient[1] + RHS.Coefficient[1];
    double c = Coefficient[2] + RHS.Coefficient[2];
    return __________________________; // line 2
}
```

24. How should the blank in line 1 be filled?
   1) void
   2) Sum
   3) Quadratic
   4) It should be left blank.
   5) None of these

25. How should the blank in line 2 be filled?
   1) Quadratic(a, b, c)
   2) Sum
   3) Quadratic(c, b, a)
   4) It should be left blank.
   5) None of these