It is dangerous to compare a signed integer to an unsigned integer because:
a) signed integers can only represent half as many values as unsigned.
b) unsigned integers cannot represent fractions to enough precision.
c) there are no negative unsigned integers.
d) type casting slows computation.

What is the maximum number of dimensions an array in C or C++ may have?
a) Two
b) Eight
c) Twenty
d) The only practical limits are memory size and compilers

What does this program print?

```
int main()
{
    int a[5] = {5, 1, 15, 20, 25};
    int i, j, m;
    i = ++a[1];
    j = a[1]++;
    m = a[i++];
    cout << i << " " << j << " " << m << endl;
    return 0;
}
```

a) 3, 2, 15
b) 2, 3, 20
c) 2, 1, 15
d) 1, 2, 5

What does this program print?

```
int main()
{
    char p;
    char buf[10] = {1, 2, 3, 4, 5, 6, 9, 8};
    p = (buf + 1)[5];
    cout << (int) p << endl;
    return 0;
}
```

a) 5
b) 6
c) 9
d) Error
e) None of the above

A standard array's elements are always stored in _____ memory locations.
a) sequential
b) random
c) sequential and random
d) data dependent
e) None of the above

The size of the array need not be specified when
a) initialization is a part of definition.
b) it is a declaration.
c) it is a formal parameter.
d) All of the above.
e) The size must always be given.

What will be printed:

```
int main()
{
    char str1[] = "abcd";
    char str2[] = "abcd";
    if (str1 == str2)
        cout << "Equal" << endl;
    else
        cout << "Unequal" << endl;
    return 0;
}
```

a) Equal
b) Unequal
c) Error
d) None of these.

What will be printed:

```
int main()
{
    string str1("abcd");
    string str2("abcd");
    if (str1 == str2)
        cout << "Equal" << endl;
    else
        cout << "Unequal" << endl;
    return 0;
}
```

a) Equal
b) Unequal
c) Error
d) None of these.

Garbage value means an unpredictable integer. What will be printed:

```
int main()
{
    int a[10];
    cout << a[-1] << " " << a[12] << endl;
    return 0;
}
```

a) 0 0
b) Garbage value 0
c) 0 Garbage Value
d) Garbage value Garbage Value
e) Code will not compile
f) Crash

Garbage value means an unpredictable integer. What will be printed:

```
int main()
{
    vector<int> a(10);
    cout << a.at(-1) << " " << a.at(12) << endl;
    return 0;
}
```

a) 0 0
b) Garbage value 0
c) 0 Garbage Value
d) Garbage value Garbage Value
e) Code will not compile
f) Crash

~~An array passed as an argument to a function is interpreted as~~
~~a) the address of the array.~~
~~b) the value of the first elements of the array.~~
~~c) the address of the first element of the array.~~
~~d) the number of elements in the array.~~

C and C++ programs are converted into machine language by
a) an editor
b) the assembler
c) the operating system
d) the preprocessor
e) None of these

C and C++ variables cannot start with
a) a number
b) special symbols other than the underscore
c) an alphabetic character
d) all of the above
e) (a) and (b)
f) (a) and (c)
g) (b) and (c)

Given a `short int` the maximum value of an unsigned integer is
a) $2^{16} – 1$
b) $2^{15} – 1$
c) 2^{16}
d) 2^{15}
e) None of these

What is the output of the program:

```
int main()
{
    int a = 11, b = 5;
    if (a = 5)
        b++;
    cout << ++a << " " << b++ << endl;
    return 0;
}
```

a) 12 7
b) 5 6
c) 6 6
d) 6 7
e) 11 6

What is the output of the program:

```
int main()
{
    int a = 3;
    for( ; a; a--)
        cout << a << " ";
    cout << endl;
    return 0;
}
```

a) no output
b) 3 2 1 0
c) 3 2 1
d) This is an infinite loop
e) Syntax error

Which command is used to skip the rest of a loop and carry on from its top?
a) break
b) resume
c) skip
d) continue
e) None of these

What will happen after attempting to compile and run the following code?

```
int main()
{
    cout << main << endl;
    return 0;
}
```

a) Will not get past the compiling step due to an error.
b) Will result in an infinite loop.
c) "main" will be printed.
d) The address of main will be printed.
e) A crash will occur.
f) None of these.

What is printed:

```
int abc(int i)
{
    return(i++);
}

int main()
{
    int i = abc(10);
    cout << --i << endl;
}
```

a) 9
b) 10
c) 11
d) None of these

How many bytes are in this struct no matter the platform?

```
struct S {
    int i;
    short s;
    char b;
};
```

a) 7
b) 8
c) 16
d) sizeof(S)
e) Cannot be determined
f) None of the above

If `bar` is a pointer to a pointer to a thing, then `&(**bar)` is:
a) a pointer to a pointer to a thing
b) a pointer to a thing
c) a thing
d) None of the above

Given the following declaration

```
char * argv[]
```

`argv` can be understood (correctly) as being:

a) a pointer to an array of char pointers
b) a pointer to pointers to char
c) the address of an array of addresses
d) (a) and (c)
e) (b) and (c)
f) (a), (b) and (c)
g) None of the above

`p` is a pointer to an integer. Which of the following increments the value of the integer?

a) p++
b) *p++
c) (*p)++
d) *(p++)
e) Cannot be determined
f) None of the above

`p` is a pointer to an integer. Which of the following increments the value of the integer after the one p points to?

a) (p + 1)++
b) *(p + 1)++
c) *(++p)++
d) *(p++)++
e) (a) and (b)
f) (b) and (c)
g) (c) and (d)
h) Cannot be determined
i) None of the above

Big Oh is a measure of the _____ of an algorithm.

a) efficiency
b) fun
c) correctness
d) clarity

A for loop with a nested while loop suggests an _____ algorithm

a) O(1)
b) O(n)
c) $O(n^2)$
d) $O(n^3)$
e) O(n log n)

The eternal battle between time and space is:

a) a great science fiction plot device
b) a way of explaining the relationship between coding time and code size
c) a way of explaining the relationship between execution time and memory / storage use
d) a way of explaining the relationship between execution time and code size
e) a way of explaining the relationship between coding time and memory / storage use

"Good style" says you should:
a) have as many returns from a function as you want
b) have as few returns from a function as practical, hopefully just one
c) never have more than one return from a function
d) (a) and (b)
e) (b) and (c)
f) (a) and (c)

Process of inserting an element in a stack is called _____
a) Create
b) Push
c) Evaluation
d) Pop

Process of removing an element from a stack is called _____
a) Create
b) Push
c) Evaluation
d) Pop

In a stack, if a user tries to remove an element from an empty stack it is called a/an _____
a) Underflow
b) Overflow
c) Delete
d) Free
e) Garbage Collection
f) None of the above

What is the value of the postfix expression `6 3 2 4 + - *`
a) Something between -5 and -15
b) Something between 5 and -5
c) Something between 5 and 15
d) Something between 15 and 25

Convert the following infix expression to Postfix form using a stack
x + y * z + (p * q + r) * s, Follow usual precedence rule and assume that the expression is legal.
a) xyz*+pq*r+s*+
b) xyz*+pq*r+s+*
c) xyz+*pq*r+s*+
d) None of the mentioned