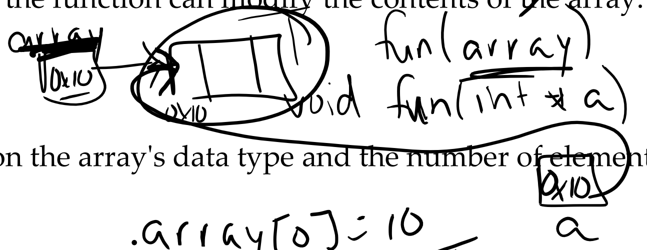
FORM 1 (Please put your name and section number (001/10am or 002/2pm) on the scantron!!!!)

CS 161 Exam II:

True (A)/False(B) (2 pts each):

1. If a function has default arguments, they can be located anywhere in the parameter list. F (int d

2. When you pass an array name as an argument to a function, the function can modify the contents of the array.



3. C++ limits the number of array dimensions to two. F

4. The amount of memory used by an array depends upon the array's data type and the number of elements in the array. T

.array[0] = 10

5. An individual array element can be processed like any other type of C++ variable/object. T
fun(array[0])

6. If you attempt to store data past an array's boundaries, it is guaranteed that the compiler will issue an error. F

int &a =

7. With reference variables you can access, but you cannot modify, data in other variables. F

8. It is legal to subtract a pointer variable from another pointer variable. T $(p1 - p2) == \emptyset$

9. All array names are pointer constants because the address stored in it cannot be changed during runtime. F
static

10. Assuming `myValues` is an array of `int` values, and `index` is an `int` variable, both of the following statements do the same thing.
*(myValues+index)

```
cout << myValues[index] << endl;
cout << *myValues + index << endl;
```

11. C++ does not perform array bounds checking, making it possible for you to assign a pointer the address of an element out of the boundaries of an array. T

12. The C++ compiler performs strict array bounds checking when it encounters an array of characters. F

13. The `strlen` function returns a C-style string's length and adds one for `\0`. F

14. You may use the == operator to compare all the elements between two C-style strings. F
w/ c++ strings   strcmp

15. Overloaded functions may have the same name, as long as their parameter lists or return types are different. F

16. A recursive function can have only one recursive case. F

17. These two declarations are exactly the same F
    char city[] = {'D', 'a', 'l', 'l', 'a', 's'};
    char city[] = "Dallas";

Multiple Choice (3 pts each):

18. These types of arguments are passed to parameters automatically if no argument is provided in the function call.
    - A) Local
    - [✓ B)] Default
    - C) Global
    - D) Relational
    - E) None of these

19. In C++, a C-string is a sequence of characters stored in consecutive memory, terminated by a ________.
    - A) period
    - B) space
    - [✓ C)] null character
    - D) semicolon
    - E) None of these

20. Which of the following function declarations is correct?
    - A) int f(int a[3][], int rowSize);
    - B) int f(int a[][], int rowSize, int columnSize);
    - [✓ C)] int f(int a[][3], int rowSize);
    - D) int f(int[][] a, int rowSize, int columnSize);

21. What is the output of the following function call, given the function definition below?

```
cout << tester (4);       // function call

int tester (int n) {      // function definition
       if (n == 1)
           return 3;
       else
           return 2 * tester ( n - 1);
}
```

1st $n\ \boxed{4}$ return 2 * 12

2nd $n\ \boxed{3}$ return 2 * 6

3rd $n\ \boxed{2}$ return 2 * 3

4th $n\ \boxed{1}$ return 3

    - A) 3
    - B) 6
    - C) 12
    - [✓ D)] 24

22. The name of an array stores the ________ of the first array element.
    - [✓ A)] memory address
    - B) value
    - C) element number
    - D) data type
    - E) None of these

23. To assign the contents of one array to another, you must use ________.
    - A) the assignment operator with the array names
    - B) the equality operator with the array names
    - [✓ C)] a loop to assign the elements of one array to the other array
    - D) Any of these
    - E) None of these

24. The function

```
int fact(int k) {
     return k*fact(k-1);
     if (k==0) return 1;
}
```

   A) computes the factorial on an integer `k` passed to it as parameter.
   B) returns the value 1 if it is passed a value of 0 for the parameter `k`.
   [✓ C)] does not correctly handle its base case.
   D) works for all non-negative values of k, but not for negative numbers.
   E) None of the above

25. What is the output of the following code?

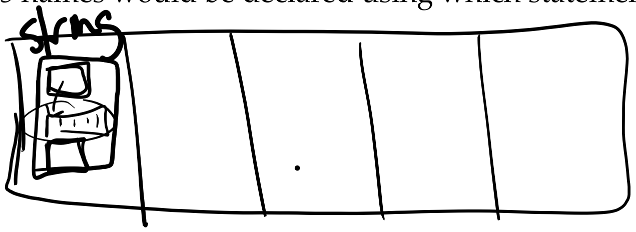
```
  #include <iostream>
  using namespace std;
  int main() {
    int matrix[3][3] = {{1, 2, 3},{4, 5, 6},{8, 9, 10}};

    int sum = 0;

    for (int i = 0; i < 3; i++)
      cout << matrix[i][1] << " ";

    return 0;
  }
```

   A) 3 6 10
   B) 1 4 8
   C) 1 2 3
   D) 4 5 6
   [✓ E)] 2 5 9

26. An array can store a group of values, but the values must be:
   [✓ A)] the same data type
   B) each of a different data type
   C) constants
   D) integers
   E) None of these

27. To pass an array as an argument to a function, pass the ________ of the array.
   A) contents
   B) size, expressed as an integer
   [✓ C)] name
   D) value of the first element
   E) None of these

28. An array of `string` objects that will hold 5 names would be declared using which statement?
   [✓ A)] `string names[5];`
   B) `string names(5);`
   C) `string names5;`
   D) `string[5] names;`
   E) None of these will work.



29. It is ________ to pass an argument to a function that contains an individual array element, such as `numbers[3]`.

A) illegal in C++
[✓ B)] legal in C++
C) not recommended by the ANSI committee
D) not good programming practice
E) None of these

30. What is the last legal subscript/index that can be used with the following array?

```
int values[5];
```

A) 0
B) 5
C) 6
[✓ D)] 4

31. Which of the following statements is not valid C++ code?

A) ~~`int ptr = &num1;`~~
[✓ B)] `int ptr = int *num1;`
C) ~~`float num1 = &ptr2;`~~
D) All of these are valid.
[✓ E)] None of these are valid.

32. What will the following code display?

0 1 2 3 4

```
int numbers[5] = { 99, 87, 66, 55, 101 };
for (int i = 1; i < 4; i++)
            cout << numbers[i] << endl;
```

A) 99
87
66
55
101

B) 87
66
55
101

[✓ C)] 87
66
55

D) Nothing. This code has an error.

33. When you work with a dereferenced pointer, you are actually working with ________.

A) a variable whose memory has been allocated
B) a copy of the value pointed to by the pointer variable
[✓ C)] the actual value/contents of the variable whose address is stored in the pointer variable
D) All of these
E) None of these

34. What will the following code do?

```
const int SIZE = 5;
double x[SIZE];
for(int i = 1; i <= SIZE; i++)
       x[i] = 0.0;
```

5 5 0 - 4 i = 0 i <= 4 i < 5

A) Each element in the array is initialized to 0.0
B) Each element in the array, except the first, is initialized to 0.0
C) Each element in the array, except the first and the last, is initialized to 0.0
[✓ D)] This code has an error that may cause it to crash.

35. What will the following code output?

```
int number = 22;
int *var = &number;
cout << var << endl;
```

0x10
22
number
0x10
var

[✓ A)] The address of the `number` variable
B) 22
C) An asterisk followed by 22
D) An asterisk followed by the address of the `number` variable

36. Which of the following statements deletes memory that has been dynamically allocated for an array?

A) `int array = delete memory;`
B) `int delete[ ];`
[✓ C)] `delete [] array;`
D) `new array = delete;`
E) `delete array [];`

37. Assuming `ptr` is a pointer variable, what will the following statement output?

```
cout << *ptr;
```

[✓ A)] The value stored in the variable whose address is contained in `ptr`.
B) The string `"*ptr"`.
C) The address of the variable stored in `ptr`.
D) The address of the variable whose address is stored in `ptr`.
E) None of these

38. Look at the following code:

```
int numbers[5] = {0, 1, 2, 3, 4 };
int *ptr = numbers;
ptr++;
```

After this code executes, which of the following statements is true?

A) `ptr` will hold the address of `numbers[0]`.
B) `ptr` will hold the address of the 2nd byte within the element `numbers[0]`.
[✓ C)] `ptr` will hold the address of `numbers[1]`.
D) This code will not compile.

39. Dynamic memory allocation occurs ________.

- ~~A)~~ when a new variable is created by the compiler
- [✓ B)] when a new variable is created at runtime
- C) when a pointer fails to dereference the right variable
- D) when a pointer is assigned an incorrect address
- E) None of these

Extra Credit (2 pts each):

40. When the less than (<) operator is used between two pointer variables, the expression is testing whether ________.

- A) the value pointed to by the first is less than the value pointed to by the second
- B) the value pointed to by the first is greater than the value pointed to by the second
- C) the address of the first variable comes before the address of the second variable in the computer's memory
- D) the first variable was declared before the second variable
- E) None of these

41. Look at the following statement:

```
sum = (*array)++;
```

This statement ________.

- A) is illegal in C++
- B) will always result in a compiler error
- C) assigns the dereferenced pointer's value, then increments the pointer's address
- D) increments the dereferenced pointer's value by one, then assigns that value
- E) None of these

42. Not all arithmetic operations may be performed on pointers. For example, you cannot ________ or ________ a pointer.

- A) multiply, divide
- B) add, subtract
- C) +=, -=
- D) increment, decrement
- E) None of these

43. True(A)/False(B) The stack frames in nested function calls are handled in a *last-in/ first-out* order.

44. True(A)/False(B) The following function will recursively add the values of each element in an array and return the sum.

```
double recSum(double array[], int count) {
   if (count > 0) {
      return recSum(array, count--) + array[count-1];
   }
}
```

F