

Student ID _____

CSE 5A

Name _____

Signature _____

Final Summer 2004

cs5a _____

Page 1 (12) _____

Page 2 (33) _____

Page 3 (32) _____

Page 4 (27) _____

Page 5 (40) _____

Page 6 (42) _____

This exam is to be taken **by yourself** with closed books, closed notes, no calculators.

Total (177 + 9 EC = 186)

Operator Precedence Table

Operators					Associativity
- (unary)	++	--	!		right to left
*	/	%			left to right
+	-				left to right
<	<=	>	>=		left to right
==	!=				left to right
&&					left to right
					left to right
=	+=	-=	*=	/=	right to left

1. Using the operator precedence table above, evaluate each expression and state what gets printed.

```
int x;
int a = 15;
int b = 10;
```

```
x = a + b % 3 * 2 - b;
printf( "%d\n", x );
```

(3 pts)

```
int x;
int a = 15;
int b = 10;
```

```
x = b + 3 - 2 * b / a;
printf( "%d\n", x );
```

(3 pts)

2. What gets printed in the following blocks of statements?

```
int a = 8;
int b = 5;
int c = -7;

if ( !(a < 6) || (b > 8) || !(c == a) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)

```
int x = -3;
int y = 10;
int z = x + 9;

if ( (z != 8) && !(x > y) && (z >= y) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)

3. Which of the following are not valid C identifiers? (Circle your answer(s).) (6 pts)

9_to_5

FM94.9

double

The_Pixies

string

Char

4. Fill in the blanks for the appropriate compilation sequence. (12 pts)

A) Executable Program

D) Linker/Linkage Editor

B) Assembler

E) C Compiler

C) C Preprocessor

F) C Source Code

_____ → _____ → _____ → _____ → _____ → _____

5. What gets printed? (15 pts)

```
void
main( void )
{
    int num = 6;

    switch ( num - 5 )
    {
        case 1:
            printf( "A\n" );
            num = num + 2;

        case 4:
            printf( "B\n" );
            num = num + 4;
            break;

        case 7:
            printf( "C\n" );
            num = num + num;

        case 6:
            printf( "D\n" );
            num = num + 5;
            break;

        default:
            printf( "E\n" );
            num = num + 8;
            break;
    }

    printf( "num = %d\n", num );
}
```

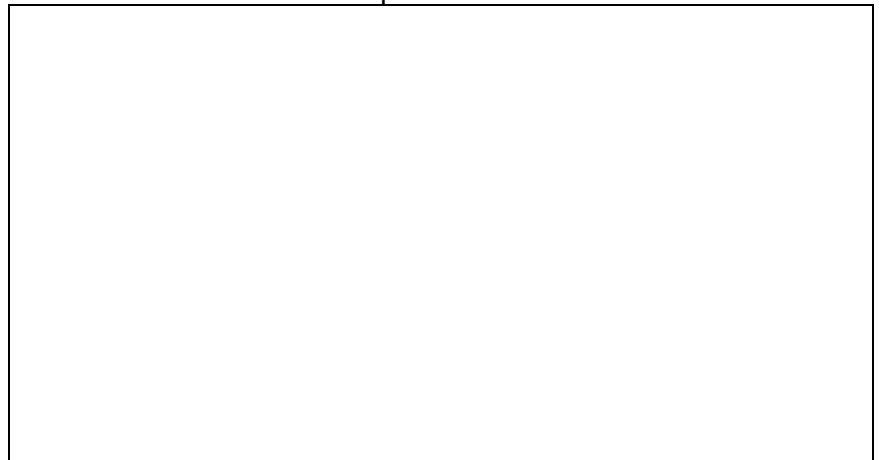
What would get printed if the switch statement read
switch(num - 4) instead of switch(num - 5)?

6. Write an equivalent **if-else** for the following **switch** statement. (12 pts)

Equivalent **if-else**

```
switch ( value )
{
    case 3:
    case 6:
        value = value * 5;
        break;

    case 19:
        value = value + 420;
        break;
}
```

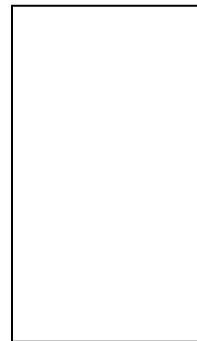


7. What gets printed in the following block of statements? (12 pts)

```
#define SIZE 8

int i;
int array[SIZE] = { 11, 2, 7, 4, 6, 3, 12, 5 };

for ( i = 0; i < SIZE; ++i )
    if ( (array[i] % 2) != 0 )
        printf( "%d\n", array[i] );
```



8. What gets printed? (8 pts)

```
#include <stdio.h>

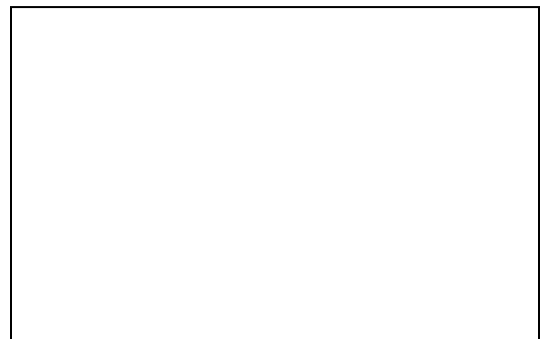
int function1( int var1, char var2 );

void
main( void )
{
    int i = 3;
    char j = '2';

    i = function1( i, j );
    printf( "%d\n", i );
}

int
function1( int var1, char var2 )
{
    int i;

    for ( i = 0; i < var1; ++i )
        printf( "%c\n", var2 );
    return ( i + var1 );
}
```



9. What gets printed? (27 pts)

```
#include <stdio.h>

#define SIZE 7

int function2( int array[], int size );

void
main( void )
{
    int array[SIZE] = { -2, 1, 4, 2, 3, -3, 6 };
    int i, result;

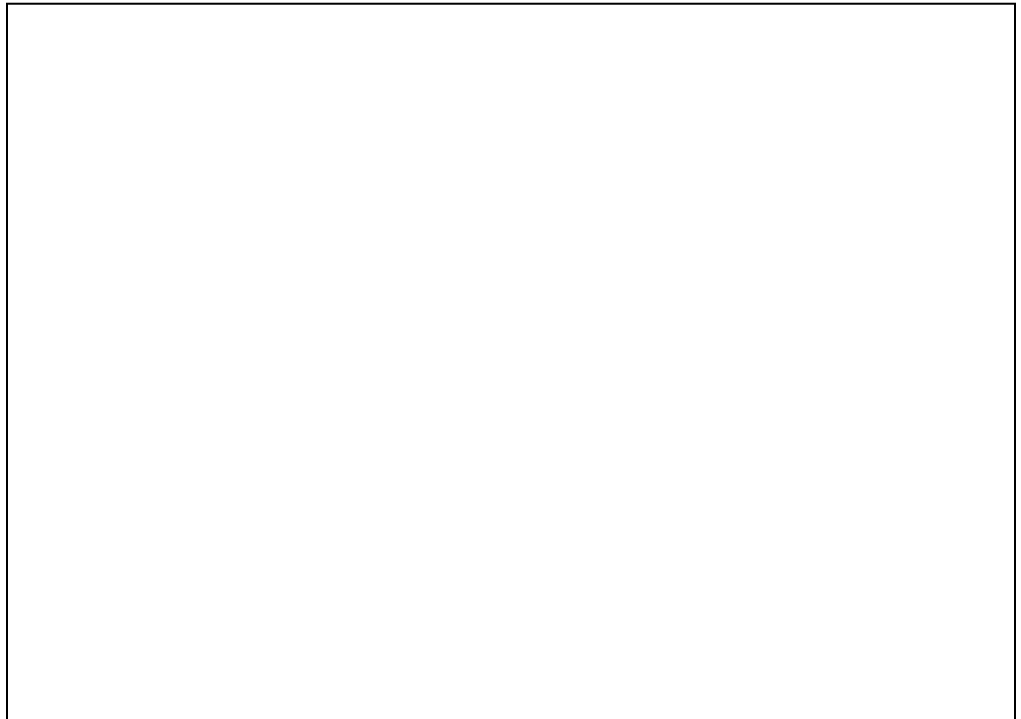
    result = function2( array, SIZE );
    printf( "Returned value = %d\n", result );

    printf( "Array elements:\n" );
    for ( i = 0; i < SIZE; ++i )
        printf( "%d\n", array[i] );
}

int
function2( int array[], int size )
{
    int i;
    int count = 0;

    for ( i = 1; i < size; ++i )
    {
        if ( array[i] >= array[i-1] )
        {
            array[i-1] = 2 * array[i];
            ++count;
        }
    }

    return count;
}
```



10. Consider the following program. Identify the marked parts, lifetime, and scope/visibility with the corresponding letter/digit from the lists below. (40 pts)

C/C++ Program Part

- A) C Preprocessor Directive
- B) Global Variable
- C) Local Variable
- D) Function Definition
- E) Internal Static Variable
- F) (Formal) Parameter
- G) Function Prototype
- H) External Static Variable

Lifetime

- 1) Entire Program
- 2) During foo() call
- 3) During func2() call

Scope/Visibility

- WW) Entire Program
- XX) This Source Module
- YY) Within foo() only
- ZZ) Within func2() only

	<u>C/C++ Program Part</u>	<u>Lifetime</u>	<u>Scope/Visibility</u>
#include <stdio.h>	_____		
#define SIZE 5	_____		
int func2(char array[]);	_____ (entire line)		
long johns = 420;	_____	_____	_____
static double ch;	_____	_____	_____
static void foo(char ch) (foo(){...})	_____	_____	_____
{	_____	_____	_____
(ch)	_____	_____	_____
int array[SIZE] = { 0, 3, 5, 2, 6 };	_____	_____	_____
static int result = 4;	_____	_____	_____
/* Other code here */	_____	_____	_____
}	_____	_____	_____
int func2(char johns[]) (func2(){...})	_____	_____	_____
{	_____	_____	_____
(johns)	_____	_____	_____
int i;	_____	_____	_____
static int result;	_____	_____	_____
/* Other code here */	_____	_____	_____
}	_____	_____	_____

How many times is the variable **result** in **foo()** initialized to 4 if **foo()** is called 8 times? _____ times

What is the initial value of the variable **result** in **func2()**? _____

How many times is the variable **result** in **func2()** given this value if **func2()** is called 8 times? _____ times

What is the initial value of the variable **i** in **func2()**? _____

How many times is the variable **array** in **foo()** initialized if **foo()** is called 8 times? _____ times

Code in **func2()** that refers to the symbol/name **ch** refers to which symbol/name?

Code in **foo()** that refers to the symbol/name **ch** refers to which symbol/name?

11. Consider the following structure definition and variable declarations. (18 pts)

```
struct Almost_Done
{
    float a;
    int b;
    int c;
    char d[8];
    int e;
};

struct Almost_Done var1, var2, var3;
```

Fill in the blanks to complete the following tasks:

```
/* Print the value of the struct member c in var2 */
printf( "%__\n", _____ );

/* Print all the elements of the struct member d in var3 */
for ( i = _____ ; i < _____ ; _____ )
    printf( "%__\n", _____ );

/* Assign the value from struct member e in var2 to the struct member b in var1 */
_____ = _____;
```

12. Consider the following strings variable definitions. (24 pts)

```
char s1[] = "Hello";
char s2[] = "Yellow?";
char s3[20];

strcpy( s3, s2 );
strcat( s3, s1 );
```

What gets printed?

```
printf( "%d", strlen( s3 ) ); _____
printf( "%s", s3 ); _____
```

Fill in the blanks to complete the following tasks:

```
/* Output "Hello ? Yellow?" in a single printf() statement. */
printf( "%__ %__ %__\n", _____, _____, _____ );

/* Change s2 to be "Yellow!" instead of "Yellow?" */
_____ = _____ ;

/* Change the 'H' in s3 to 'h' using the tolower() function */
_____ = _____ ;
```

Scratch Paper

Scratch Paper