

Student ID \_\_\_\_\_

# CSE 5A Midterm Fall 2003

Name \_\_\_\_\_

Signature \_\_\_\_\_

cs5a \_\_\_\_\_

Page 1 (12) \_\_\_\_\_

Page 2 (21) \_\_\_\_\_

Page 3 (30) \_\_\_\_\_

Page 4 (21) \_\_\_\_\_

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

**Operator Precedence Table**

Total (84) \_\_\_\_\_

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 = 10;
int b = 15;
```

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

(3 pts)

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

```
x = b * 2 / 3 - 4 + a;
printf( "%d\n", x );
```

(3 pts)

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

```
int a = 5;
int b = 8;
int c = 15;

if ( ( a > 6 ) || ( b <= 8 ) || ( c == 19 ) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)

```
int x = -3;
int y = 0;
int z = x + 9;

if ( ( z == 6 ) && ( x < y ) && ( z < x ) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)

3. Which of the following are valid C identifiers? (Circle your answer(s).) (6 pts)

\_undeR\_scorE

aTdHvAaNnKcSe

1\_4\_the\_Road

XTC\_4\_Me

D&B

Super\*

4. Fill in the blanks with the appropriate format specifiers to output the values correctly. (6 pts)

```
void
main( void )
{
    int    a = 5;
    float  b = 10.5;
    char   c = 'K';

    printf( "a = %_____ \nb= %_____ \nc= %_____" , a, b, c );
}
```

5. What gets printed? (9 pts)

```
void
main( void )
{
    int num = 1;

    switch ( num + 3 )
    {
        case 1:
            printf( "A\n" );
            num = num + 2;

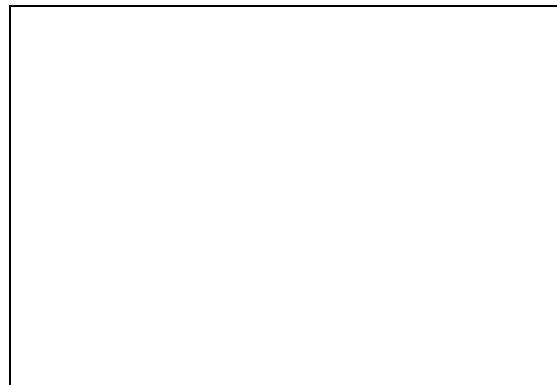
        case 4:
            printf( "B\n" );
            num = num + 4;

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

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

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

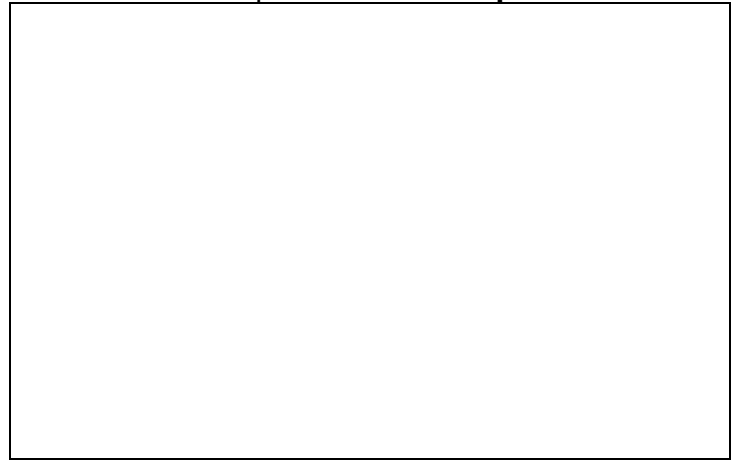
    printf( "num = %d\n", num );
}
```



6. Write an equivalent **while loop** for the following **for loop**. (12 pts)

```
for ( i = 0; i < j; ++i )
    printf( "%d %d\n", i, j - i );
```

Equivalent **while loop**

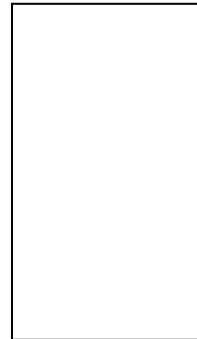


7. What gets printed in the following block of statements? (12 pts)

```
#define SIZE 8

int i;
int array[SIZE] = { 11, 7, 3, 4, 6, 2, -1, 5 };

for ( i = 0; i < SIZE; ++i )
    if ( array[i] > 4 )
        printf( "%d\n", array[i] );
```



8. What gets printed? (6 pts)

```
#include <stdio.h>

void function1( int var1, char var2 );

void
main( void )
{
    int i = 3;
    char j = '8';

    function1( i, j );
}

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

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



9. What gets printed? (12 pts)

```
#include <stdio.h>

#define SIZE 5

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

void
main( void )
{
    int array[SIZE] = { 4, 1, 0, 2, 3 };
    int result;

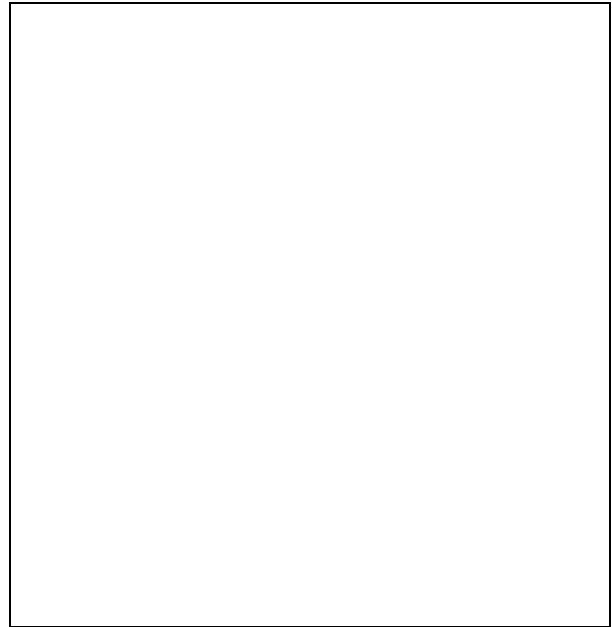
    result = function2( array, SIZE );

    printf( "%d\n", result );
}

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

    for ( i = 0; i < size; ++i )
    {
        printf( "%d\n", result );
        result = result + array[i];
    }

    return result;
}
```



Use the following to answer the questions below:

- A) Function Prototype
- B) Local Variable
- C) Return Type
- D) Function Definition
- E) Parameter

What is `i` in `function2()`? (3 pts)

What is `size` in `function2()`? (3 pts)

What is the `int` in front of `function2()`? (3 pts)