Name:______________________________ NetID:____________

Answer all questions in the space provided. Write clearly and legibly, you will not get credit for illegible or incomprehensible answers. Print your name at the top of every page.

This is a closed book exam. However, each student is allowed to bring one page of notes to the exam. Also, you are permitted the use of a "dumb" calculator to perform basic arithmetic.

Question:	1	2	3	4	5	6	7	8	9	Total
Points:	12	15	12	12	12	5	6	6	10	90
Score:										

1. For these questions, select all that apply.

   (a) Which of the following expressions are considered to be true values in the C programming language? (6)

   A. `0`
   B. `'\0'`
   C. `'0'`
   D. `1`
   E. `'1'`
   F. `-1`
   G. `3 >> 2`
   H. `3 << 2`
   I. `3 < 2`
   J. `0.75`
   K. `3 / 4`
   L. `3.0 / 4.0`

   (b) Which of the following are keywords in the C programming language? (6)

   A. boolean
   B. do
   C. until
   D. for
   E. if
   F. then
   G. int
   H. integer
   I. while
   J. import
   K. include
   L. case

2. For these questions, select the single best answer.

   (a) Assume `x` is already declared and initialized and consider the following code snippet. (3)

```
  if ( x = 0 )
    printf ( "x is zero\n" );
```

   A. This is invalid code because of the use of the assignment operator in the if test.
   B. This is invalid code because `printf` is only given one argument.
   C. This is valid code. It prints “x is zero”
   D. This is valid code, but nothing will be printed.

   (b) What is the final value of `x` when the following code is run? (3)

```
int x;
for(x=10; x>5; --x) printf("%d", x);
```

   A. 0
   B. 1
   C. 4
   D. 5
   E. 6
   F. 10
   G. undefined

   (c) What is the return type of the function with the following prototype? (3)

```
int foo(char x, float y, double z);
```

   A. char
   B. double
   C. float
   D. foo
   E. int
   F. void

   (d) Which of the following gives the memory address of a variable `x`? (3)

   A. `x`
   B. `addressof(x)`
   C. `*x`
   D. `&x`
   E. `~x`

   (e) Which of the following gives the value stored at the address pointed to by a pointer `p`? (3)

   A. `p`
   B. `valueof(p)`
   C. `*p`
   D. `&p`
   E. `~p`

3. The following program compiles and runs. What is its output? (12)

```
#include <stdio.h>

int n = 16;

int foo(int x)
{
  int y = n / x;
  x += y;
  n -= y;
  printf("foo: x=%d, y=%d, n=%d\n", x, y, n);
  return y;
}

int main()
{
  int x, n;
  n = 5;
  x = foo(n);
  printf("main: n=%d, x=%d\n", n, x);

  n = foo(x);
  printf("main: n=%d, x=%d\n", n, x);

  return 0;
}
```

4. The following program compiles and runs. What is its output? (12)

```c
#include <stdio.h>

int main()
{
  unsigned char x = 45;
  unsigned char y = 30;
  unsigned char z = 238;

  unsigned char a = x << 3;
  unsigned char b = x >> 2;
  unsigned char c = x & y;
  unsigned char d = y & z;
  unsigned char e = x | y;
  unsigned char f = x ^ y;

  printf("a = %d\n", a);
  printf("b = %d\n", b);
  printf("c = %d\n", c);
  printf("d = %d\n", d);
  printf("e = %d\n", e);
  printf("f = %d\n", f);

  return 0;
}
```

5. What is the output of this program? (12)

```
#include <stdio.h>

int searchRange(int x, int vals[], int low, int high)
{
  if(low <= high)
  {
    int mid = (low+high) / 2;

    printf("[%d %d %d] ", low, mid, high);

    if (x < vals[mid]) return searchRange(x, vals, low, mid-1);
    else if (x > vals[mid]) return searchRange(x, vals, mid+1, high);
    else return mid;
  }
  else return -1;
}



int main()
{
  int nums[] = {12, 13, 15, 17, 21, 23, 27, 39, 43, 51};
  printf("index = %d\n", searchRange(14, nums, 0, 9));
  printf("index = %d\n", searchRange(23, nums, 0, 9));
  return 0;
}
```

6. I have declared two `unsigned char` variables, `a` and `b`. Assume that `a` has been initialized. (5)

   I would like to assign a value to `b` such that the lowest four bits of `a` are the highest four bits of `b` and the highest four bits of `a` are the lowest four bits of `b`.

   Write a single line of C code that will accomplish this.

7. The following program compiles and runs. What is its output? (6)

```
#include <stdio.h>

int main()
{
  char data[] = "SPLINT";
  char *linePt = &data[2];
  *linePt = ’R’;
  data[5] = ’G’;
  printf("[%s], [%s]\n", data, linePt);

  return 0;
}
```

8. The following program compiles and runs. What is its output? (6)

```c
#include <stdio.h>

struct Point { int x; int y; };

struct Point foo(struct Point p1, struct Point *p2)
{
  p1.x += p2->x;
  p2->y += p1.y;
  return p1;
}

int main()
{
  struct Point a = {1, 2};
  struct Point b = {3, 4};
  struct Point c = foo(a, &b);

  printf("a=(%d, %d)\n", a.x, a.y);
  printf("b=(%d, %d)\n", b.x, b.y);
  printf("c=(%d, %d)\n", c.x, c.y);

  return 0;
}
```

9. The following program is meant to print out the factors of a number. However, the programmer made mistakes that prevent the program from compiling and running properly. (10)

   - Which lines need to be changed to fix this program?
   - Why do they need to be changed? (What is wrong with them?)
   - How do you change the problem lines to fix the program?

```
 1 #include <stdio.h>
 2
 3 void printFactors(float n);
 4
 5 int main()
 6 {
 7   printFactors(36);
 8   return 0;
 9 }
10
11 void printFactors(int n)
12 {
13   int i = 2;
14   while(i < n);
15   {
16     if(n % i == 0) printf("%d ", i);
17     ++i;
18   }
19   printf("\n");
20 }
```