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 | Total |
|-----------|---|---|---|---|---|---|---|-------|
| Points: | 10 | 12 | 6 | 12 | 12 | 12 | 6 | 70 |
| Score: | | | | | | | | |

1. For these questions, select all that apply.

   (a) Which of the following are *not* keywords in the C programming language?     (5)

   A. boolean
   B. case
   C. do
   D. pointer
   E. struct
   F. type
   G. typedef
   H. union
   I. unless
   J. while

   (b) Which of the following are considered to be true values in the C programming language?     (5)

   A. `0`
   B. `'\0'`
   C. `'0'`
   D. `1`
   E. `'1'`
   F. `-1`
   G. `sizeof(char)-1`
   H. `8 >> 4`
   I. `0.25`
   J. `2/3`
   K. `2.0/3.0`

2. Give a short answer to each of the following questions.

    (a) What is the operator >>= used for?  Write an expression that uses it and show another     (4)
        way to write the same expression.

    (b) Why is it important to always initialize your variables in a C program?     (4)

    (c) Assuming x has been previously declared and initialized — Is the following code snippet     (4)
        valid C source code? If not, why not? If so, what is the result when this code is executed?

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

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

```
1  #include <stdio.h>
2
3  int main()
4  {
5     char data[] = "MidtermTime";
6     char *linePt = &data[3];
7     data[8] = '\0';
8     *(linePt+4) = '!';
9     printf("[%s], [%s]\n", data, linePt);
10
11    return 0;
12 }
```

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

```c
#include <stdio.h>

int x=1;

int foo(int n)
{
  int y=7;
  x *= 2;
  y -= x;
  n += y;
  printf("foo: x=%d, y=%d, n=%d\n", x, y, n);
  return n;
}

int main()
{
  int x, n;
  n = 3;
  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;
}
```

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

```c
#include <stdio.h>

void main(void)
{
  char s[] = "XcsXXfXunX";
  char del = 'X';

  int sourceIndex = 0;
  int sinkIndex = 0;
  while (s[sourceIndex])
  {
    if (s[sourceIndex] != del)
    {
      s[sinkIndex] = s[sourceIndex];
      sinkIndex++;
    }
    else
    {
      printf("[%d,%d] %s\n", sourceIndex, sinkIndex, s);
    }
    sourceIndex++;
  }
  s[sinkIndex]='\0';
  printf("result: %s\n",s);
}
```

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

```c
#include <stdio.h>

int main()
{
  unsigned char x = 27;
  unsigned char y = 41;
  unsigned char z = 175;

  unsigned char a = x >> 2;
  unsigned char b = x << 4;
  unsigned char c = x & y;
  unsigned char d = x & 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;
}
```

7. 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;
   p1.y += p2->y;
   p2->x += p1.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;
}
```