For each program, tell me the output of the program. Put answers on the answer sheet.

### Program 1
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

main()
{
    char x[41], \*y;
    strcpy(x, "0123456789012345678901234567890123456789");
    for (y = x; y < x+41; y += 7) \*y = '\0';
    for (y = x; y < x+41; y += 5) printf("(%s)\n", y);
}
```

### Program 2
```c
main()
{
    int i, j, k;
    i = 0xd00f;
    j = 0x7801;
    k = 0x7452;
    printf("1: 0x%x\n", (i >> 1));
    printf("2: 0x%x\n", (j << 2));
    printf("3: 0x%x\n", (i << 4) | j);
    printf("4: 0x%x\n", (k >> 12) & i);
    i *= i;
    k *= j;
    i += j;
    printf("5: 0x%x\n", (i ^ j ^ i));
}
```

### Program 3
```c
typedef unsigned int UI;

int pm(char \*a)
{
    char \*s1;
    char **s2;
    unsigned int i;
    s1 = a + 21;
    printf("1: %s\n", s1);
    s2 = (char **) a;
    s2 += 4;
    printf("2: %s\n", \*s2);
    s1 = a + 8;
    s2 = (char **) s1;
    printf("3: 0x%08x\n", (UI) \*s2);
    a[0] = 1;
    memcpy(&i, a, 4);
    printf("4: 0x%08x\n", i);
}
```

### Program 4
State of memory when pm() is called in program 4.
The value of a is 0xbfffda60.

<table>
<thead>
<tr>
<th>Address</th>
<th>Contents-hex</th>
<th>Contents in chars</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xbfffda60</td>
<td>0x79206f73</td>
<td>'s'</td>
</tr>
<tr>
<td>0xbfffda64</td>
<td>0xbfffda8c</td>
<td>--</td>
</tr>
<tr>
<td>0xbfffda68</td>
<td>0x20616e6e</td>
<td>'n'</td>
</tr>
<tr>
<td>0xbfffda6c</td>
<td>0x706d7564</td>
<td>'d'</td>
</tr>
<tr>
<td>0xbfffda70</td>
<td>0xbfffda84</td>
<td>--</td>
</tr>
<tr>
<td>0xbfffda74</td>
<td>0x276f7920</td>
<td>' '</td>
</tr>
<tr>
<td>0xbfffda78</td>
<td>0x69727420</td>
<td>' '</td>
</tr>
<tr>
<td>0xbfffda7c</td>
<td>0x62206b63</td>
<td>'c'</td>
</tr>
<tr>
<td>0xbfffda80</td>
<td>0x002e6761</td>
<td>'a'</td>
</tr>
<tr>
<td>0xbfffda84</td>
<td>0x65736165</td>
<td>'e'</td>
</tr>
<tr>
<td>0xbfffda88</td>
<td>0x006e6f20</td>
<td>--</td>
</tr>
<tr>
<td>0xbfffda8c</td>
<td>0x61206e69</td>
<td>'i'</td>
</tr>
<tr>
<td>0xbfffda90</td>
<td>0x70696820</td>
<td>' '</td>
</tr>
<tr>
<td>0xbfffda94</td>
<td>0x61687420</td>
<td>' '</td>
</tr>
<tr>
<td>0xbfffda98</td>
<td>0x002e6766e</td>
<td>'n'</td>
</tr>
</tbody>
</table>

For programs 4 - 7, I will give you a procedure pm(), and a state of memory, and I will tell you the value of the parameter to the procedure pm(). Assume that pointers are 4 bytes.

First off, there are no seg faults in this test. If you tell me that there is a seg fault, you are simply losing points. I have compiled and executed each of these programs.

Second, please do your work and write your answers on scratch paper, and then copy them to the answer sheet. I am grading exact output here, so it should be neat.

When I give you the contents of memory, I will give them to you in potentially three ways – in decimal, in hexadecimal, and as four chars. When I give them as chars, I show printable chars and the null character. If the character is unprintable, I specify that with "--".

For example, in the memory specification to the left for Program 4, the character 's' is at address 0xbfffda60, and the character 'o' is at 0xbfffda61. The character at address 0xbfffda64 is unprintable. It is the byte 0x8e.

strncpy(s1, s2) copies s2 to s1.
strchr(s1, c) returns a pointer to the first occurrence of c in s1.
Program 5

```c
int pm(int *p)
{
    int **q;
    int ***r;
    int ****s;

    q = (int **) p;
    r = (int ***) p;
    s = (int ****) p;

    printf("1: %d\n", *p);
    printf("2: %d\n", **q);
    printf("3: %d\n", ***r);
    printf("4: %d\n", ****s);
    printf("5: %d\n", p[1]);
    printf("6: %d\n", q[1][1]);
    printf("7: %d\n", r[1][1][1]);
    printf("8: %d\n", s[1][1][1][1]);
}
```

Program 6

```c
int pm(int *p)
{
    char *s;
    char **t;

    s = (char *) p[3];
    printf("1. %s\n", s);
    s = (char *) p;
    s[0]++;
    s[3] = 'T';
    printf("2. %s\n", s);
    printf("3. 0x%08x\n", *p);
    printf("4. 0x%08x\n", p[1] | p[2]);
    printf("5. 0x%08x\n", p[3] ^ p[4]);

    t = (char **) &(p[5]);
    printf("6. 0x%08x\n", (unsigned int) (*t));
    printf("7. %s\n", *t);
}
```

Program 7

typedef unsigned int UI;

typedef struct X {
    int *ip;
    char *jp;
    struct X *kp;
    int x;
} XS;

```c
int pm(int *p)
{
    XS *z;

    z = (XS *) p;

    printf("1: %d\n", z->x);
    printf("2: %s\n", z->jp);
    printf("3: %d\n", *(z->ip));
    printf("4: %d\n", z->ip[1]);

    z = z->kp;

    printf("5: 0x%08x\n", (UI) &z->x);
    printf("6: %s\n", z->jp);
    printf("7: %d\n", *(z->ip));
    printf("8: %d\n", *(z->kp->ip));
}
```
Program 1
1:
2:
3:
4:
5:

Program 2

Program 3

Program 4
1:
2:
3:
4:

Program 5
1:
2:
3:
4:
5:
6:
7:
8:

Program 6
1:
2:
3:
4:
5:
6:
7:
8:

Program 7
1:
2:
3:
4:
5:
6:
7:
8:

Name: ______________________
UTK Email: __________________