CS360 Midterm 1 - February 21, 2017 - James S. Plank

Put all answers on the answer sheet. In all of these questions, please assume the following:

- Pointers and longs are 4 bytes.
- The machine is little endian (like our lab machines and my Mac). So, if an integer is 0xabcdef88, then its first byte is 0x88, and its last byte is 0xab.
- There are no segmentation violations or bus errors in any of this code.

Here are prototypes of **strcpy()**, **strcat()** and **memcpy()**:

```
char *strcpy(char *to, char *from);
char *strcat(char *to, char *from);
void *memcpy(void *to, void *from, int number_of_bytes);
```

Question 1

What is the output of the program below:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
  unsigned char c, d;
  unsigned int i, j, k, n, m;

  i = 0x57446812;       j = 0x78210735;

  k = 0x0f0f0f0f;       n = 0x57440000;

  m = 0x6812;

  c = (i & 0xff);
  d = (c << 4);

  printf("1: 0x%08x\n", (n << 3));
  printf("2: 0x%08x\n", (m >> 2));
  printf("3: 0x%08x\n", i & k);
  printf("4: 0x%08x\n", i & (k << 4));
  printf("5: 0x%08x\n", j | k);
  printf("6: 0x%08x\n", i >> 16);
  printf("7: 0x%08x\n", j << 12);
  printf("8: 0x%08x\n", c);
  printf("9: 0x%08x\n", d);
  printf("A: 0x%08x\n", i ^ n);
  printf("B: 0x%08x\n", n + 16);
}
```

Question 2

What is the output of the program below:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
  char buf[50];
  char *x, *y, *z;

  x = buf;
  y = buf+10;
  z = buf+20;

  strcpy(buf, "0123456789ABCDEFGHIJ0123456789");
  strcpy(z, "-");
  strcat(y, "#");
  *y = '\0';
  z[5] = '\0';

  for (x = buf; x - buf < 35; x += 7) {
    printf("%s\n", x);
  }
}
```

Work Space

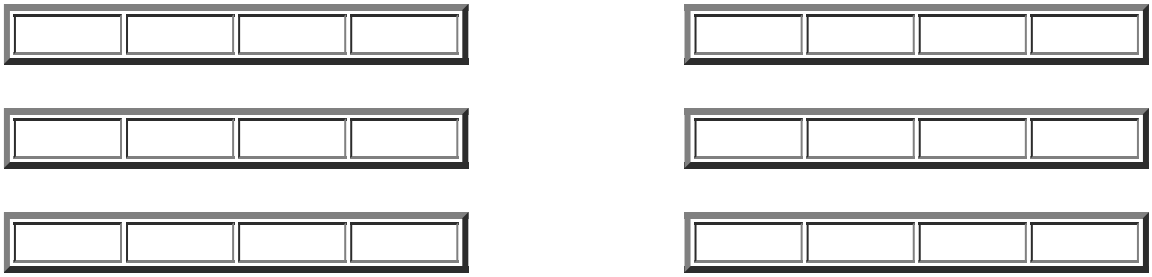
0123456789ABCDEFGHIJ0123456789

Question 3

What is the output of the following program?

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main()
{
  unsigned int i[2];
  unsigned char *c, *vp;
  int v;

  i[0] = 0xabcdef67;
  i[1] = 0x13da24eb;

  vp = (unsigned char *) &v;
  c = (unsigned char *) i;

  printf("1: 0x%02x 0x%02x 0x%02x 0x%02x\n", c[0], c[1], c[2], c[3]);
  printf("2: 0x%02x 0x%02x 0x%02x 0x%02x\n", c[4], c[5], c[6], c[7]);

  memcpy(&v, c+2, 4);
  printf("3: 0x%08x\n", v);

  memcpy(vp, c, 2);
  memcpy(vp+2, c+6, 2);
  printf("4: 0x%08x\n", v);

  i[0] += 0x10101;
  printf("5: 0x%02x 0x%02x 0x%02x 0x%02x\n", c[0], c[1], c[2], c[3]);

  /* These last two will be treated as extra credit.
     If you're running behind, skip them.  */

  c[4]++;
  c[5] += 2;

  printf("6: 0x%08x\n", i[0]);
  printf("7: 0x%08x\n", i[1]);

}
```

Workspace



Question 4

When the following procedure is called, the value of **x** is `0x78e51120`, and the contents of the 48 bytes starting with **x** are as follows. You'll note, I'm printing each set of eight bytes as a double, and each set of four bytes in decimal, in hexadecimal, and as four characters. Where I print the characters, unprintable characters are printed with "--".

Address	8 bytes printed as a double using printf("%14.8le",	4 bytes printed using printf("%d",	4 bytes printed using printf("0x%x",	4 bytes printed as four characters
0x78e51120	2.27938697e+274	2028278064	0x78e51130	'0' | -- | -- |'x'
0x78e51124		2028278084	0x78e51144	'D' | -- | -- |'x'
0x78e51128	2.27936055e+274	2028278088	0x78e51148	'H' | -- | -- |'x'
0x78e5112c		2028278068	0x78e51134	'4' | -- | -- |'x'
0x78e51130	2.27934735e+274	2028278080	0x78e51140	'@' | -- | -- |'x'
0x78e51134		2028278060	0x78e5112c	',' | -- | -- |'x'
0x78e51138	2.27938697e+274	2028278064	0x78e51130	'0' | -- | -- |'x'
0x78e5113c		2028278084	0x78e51144	'D' | -- | -- |'x'
0x78e51140	2.27940017e+274	2028278072	0x78e51138	'8' | -- | -- |'x'
0x78e51144		2028278092	0x78e5114c	'L' | -- | -- |'x'
0x78e51148	2.27939357e+274	2028278072	0x78e51138	'8' | -- | -- |'x'
0x78e5114c		2028278088	0x78e51148	'H' | -- | -- |'x'

Your job is to tell me the output of the following program:

```
void a(char *x)
{
  char       **y, ***z;
  int    *i, **j, ***k;
  double *d, **e, ***f;

  y = (char **) x;
  z = (char ***) x;

  printf("1: %c\n", *x);
  printf("2: %c\n", **y);
  printf("3: %c\n", ***z);

  i = (int *) x;
  i++;
  j = (int **) i;
  k = (int ***) i;

  printf("4: %d\n", i[1]);
  printf("5: %d\n", **j);
  printf("6: %d\n", ***k);

  d = (double *) x;
  d++;
  e = (double **) d;
  f = (double ***) d;

  printf("7: %14.8le\n", d[1]);
  printf("8: %14.8le\n", **e);
  printf("9: %14.8le\n", ***f);
  printf("A: %14.8le\n", e[1][1]);
}
```

Question 5

When the procedure to the right is called, the value of **p** is `0x61626324`, and the contents of the 92 bytes starting with **p** are as shown below. Each set of four bytes is printed in decimal, hexadecimal, and as four characters. You'll note that this time, all of the characters are printable.

Address	Decimal	Hexadecimal	As four characters
0x61626324	1633837882	0x6162633a	':' |'c' |'b' |'a'
0x61626328	1633837928	0x61626368	'h' |'c' |'b' |'a'
0x6162632c	1633837875	0x61626333	'3' |'c' |'b' |'a'
0x61626330	1633837868	0x6162632c	',' |'c' |'b' |'a'
0x61626334	1633837870	0x6162632e	'.' |'c' |'b' |'a'
0x61626338	1633837934	0x6162636e	'n' |'c' |'b' |'a'
0x6162633c	1633837934	0x6162636e	'n' |'c' |'b' |'a'
0x61626340	1633837904	0x61626350	'P' |'c' |'b' |'a'
0x61626344	1633837887	0x6162633f	'?' |'c' |'b' |'a'
0x61626348	1633837928	0x61626368	'h' |'c' |'b' |'a'
0x6162634c	1633837863	0x61626327	''' |'c' |'b' |'a'
0x61626350	1633837872	0x61626330	'0' |'c' |'b' |'a'
0x61626354	1633837909	0x61626355	'U' |'c' |'b' |'a'
0x61626358	1633837933	0x6162636d	'm' |'c' |'b' |'a'
0x6162635c	1633837860	0x61626324	'$' |'c' |'b' |'a'
0x61626360	1633837880	0x61626338	'8' |'c' |'b' |'a'
0x61626364	1633837879	0x61626337	'7' |'c' |'b' |'a'
0x61626368	1633837890	0x61626342	'B' |'c' |'b' |'a'
0x6162636c	1633837903	0x6162634f	'O' |'c' |'b' |'a'
0x61626370	1633837900	0x6162634c	'L' |'c' |'b' |'a'
0x61626374	1633837906	0x61626352	'R' |'c' |'b' |'a'
0x61626378	1633837944	0x61626378	'x' |'c' |'b' |'a'
0x6162637c	1633837927	0x61626367	'g' |'c' |'b' |'a'

Tell me the output of the procedure.

```
typedef struct ms {
   char a;
   char b;
   char c;
   char d;
   int e;
   char *f;
   struct ms *g;
} Mystruct;

void pm(Mystruct *p)
{
  Mystruct *q;

  printf("1: 0x%x\n", (unsigned int) p);
  printf("2: %c %c %c %c\n", p->a, p->b, p->c, p->d);
  printf("3: %d\n", p->e);
  printf("4: %c %c %c %c\n", p->f[0], p->f[1], p->f[2], p->f[3]);
  printf("\n");

  printf("5: 0x%x\n", (unsigned int) &(p[1]));
  printf("6: %c %c %c %c\n", p[1].a, p[1].b, p[1].c, p[1].d);
  printf("7: %d\n", p[1].e);
  printf("8: %c %c %c %c\n", p[1].f[0], p[1].f[1], p[1].f[2], p[1].f[3]);
  printf("\n");

  q = p+4;

  printf("9: 0x%x\n", (unsigned int) q);
  printf("A: %c %c %c %c\n", q->a, q->b, q->c, q->d);
  printf("B: %d\n", q->e);
  printf("C: %c %c %c %c\n", q->f[0], q->f[1], q->f[2], q->f[3]);
  printf("\n");

  q = p->g;

  printf("D: 0x%x\n", (unsigned int) q);
  printf("E: %c %c %c %c\n", q->a, q->b, q->c, q->d);
  printf("F: %d\n", q->e);
  printf("G: %c %c %c %c\n", q->f[0], q->f[1], q->f[2], q->f[3]);
  printf("\n");

  q = q[1].g;

  printf("H: 0x%x\n", (unsigned int) q);
  printf("I: %c %c %c %c\n", q->a, q->b, q->c, q->d);
  printf("J: %d\n", q->e);
  printf("K: %c %c %c %c\n", q->f[0], q->f[1], q->f[2], q->f[3]);
  printf("\n");
}
```

Work Sheet for Question 4

Address	8 bytes printed as a double using printf("%14.8le",	4 bytes printed using printf("%d",	4 bytes printed using printf("0x%x",	4 bytes printed as four characters
0x78e51120	2.27938697e+274	2028278064	0x78e51130	'0' | -- | -- |'x'
0x78e51124		2028278084	0x78e51144	'D' | -- | -- |'x'
0x78e51128	2.27936055e+274	2028278088	0x78e51148	'H' | -- | -- |'x'
0x78e5112c		2028278068	0x78e51134	'4' | -- | -- |'x'
0x78e51130	2.27934735e+274	2028278080	0x78e51140	'@' | -- | -- |'x'
0x78e51134		2028278060	0x78e5112c	',' | -- | -- |'x'
0x78e51138	2.27938697e+274	2028278064	0x78e51130	'0' | -- | -- |'x'
0x78e5113c		2028278084	0x78e51144	'D' | -- | -- |'x'
0x78e51140	2.27940017e+274	2028278072	0x78e51138	'8' | -- | -- |'x'
0x78e51144		2028278092	0x78e5114c	'L' | -- | -- |'x'
0x78e51148	2.27939357e+274	2028278072	0x78e51138	'8' | -- | -- |'x'
0x78e5114c		2028278088	0x78e51148	'H' | -- | -- |'x'

Address	8 bytes printed as a double using printf("%14.8le",	4 bytes printed using printf("%d",	4 bytes printed using printf("0x%x",	4 bytes printed as four characters
0x78e51120	2.27938697e+274	2028278064	0x78e51130	'0' | -- | -- |'x'
0x78e51124		2028278084	0x78e51144	'D' | -- | -- |'x'
0x78e51128	2.27936055e+274	2028278088	0x78e51148	'H' | -- | -- |'x'
0x78e5112c		2028278068	0x78e51134	'4' | -- | -- |'x'
0x78e51130	2.27934735e+274	2028278080	0x78e51140	'@' | -- | -- |'x'
0x78e51134		2028278060	0x78e5112c	',' | -- | -- |'x'
0x78e51138	2.27938697e+274	2028278064	0x78e51130	'0' | -- | -- |'x'
0x78e5113c		2028278084	0x78e51144	'D' | -- | -- |'x'
0x78e51140	2.27940017e+274	2028278072	0x78e51138	'8' | -- | -- |'x'
0x78e51144		2028278092	0x78e5114c	'L' | -- | -- |'x'
0x78e51148	2.27939357e+274	2028278072	0x78e51138	'8' | -- | -- |'x'
0x78e5114c		2028278088	0x78e51148	'H' | -- | -- |'x'

Work Sheet for Question 5

Address	Decimal	Hexadecimal	As four characters
0x61626324	1633837882	0x6162633a	':' |'c' |'b' |'a'
0x61626328	1633837928	0x61626368	'h' |'c' |'b' |'a'
0x6162632c	1633837875	0x61626333	'3' |'c' |'b' |'a'
0x61626330	1633837868	0x6162632c	',' |'c' |'b' |'a'
0x61626334	1633837870	0x6162632e	'.' |'c' |'b' |'a'
0x61626338	1633837934	0x6162636e	'n' |'c' |'b' |'a'
0x6162633c	1633837934	0x6162636e	'n' |'c' |'b' |'a'
0x61626340	1633837904	0x61626350	'P' |'c' |'b' |'a'
0x61626344	1633837887	0x6162633f	'?' |'c' |'b' |'a'
0x61626348	1633837928	0x61626368	'h' |'c' |'b' |'a'
0x6162634c	1633837863	0x61626327	''' |'c' |'b' |'a'
0x61626350	1633837872	0x61626330	'0' |'c' |'b' |'a'
0x61626354	1633837909	0x61626355	'U' |'c' |'b' |'a'
0x61626358	1633837933	0x6162636d	'm' |'c' |'b' |'a'
0x6162635c	1633837860	0x61626324	'$' |'c' |'b' |'a'
0x61626360	1633837880	0x61626338	'8' |'c' |'b' |'a'
0x61626364	1633837879	0x61626337	'7' |'c' |'b' |'a'
0x61626368	1633837890	0x61626342	'B' |'c' |'b' |'a'
0x6162636c	1633837903	0x6162634f	'O' |'c' |'b' |'a'
0x61626370	1633837900	0x6162634c	'L' |'c' |'b' |'a'
0x61626374	1633837906	0x61626352	'R' |'c' |'b' |'a'
0x61626378	1633837944	0x61626378	'x' |'c' |'b' |'a'
0x6162637c	1633837927	0x61626367	'g' |'c' |'b' |'a'

0x61626324	1633837882	0x6162633a	':' |'c' |'b' |'a'
0x61626328	1633837928	0x61626368	'h' |'c' |'b' |'a'
0x6162632c	1633837875	0x61626333	'3' |'c' |'b' |'a'
0x61626330	1633837868	0x6162632c	',' |'c' |'b' |'a'
0x61626334	1633837870	0x6162632e	'.' |'c' |'b' |'a'
0x61626338	1633837934	0x6162636e	'n' |'c' |'b' |'a'
0x6162633c	1633837934	0x6162636e	'n' |'c' |'b' |'a'
0x61626340	1633837904	0x61626350	'P' |'c' |'b' |'a'
0x61626344	1633837887	0x6162633f	'?' |'c' |'b' |'a'
0x61626348	1633837928	0x61626368	'h' |'c' |'b' |'a'
0x6162634c	1633837863	0x61626327	''' |'c' |'b' |'a'
0x61626350	1633837872	0x61626330	'0' |'c' |'b' |'a'
0x61626354	1633837909	0x61626355	'U' |'c' |'b' |'a'
0x61626358	1633837933	0x6162636d	'm' |'c' |'b' |'a'
0x6162635c	1633837860	0x61626324	'$' |'c' |'b' |'a'
0x61626360	1633837880	0x61626338	'8' |'c' |'b' |'a'
0x61626364	1633837879	0x61626337	'7' |'c' |'b' |'a'
0x61626368	1633837890	0x61626342	'B' |'c' |'b' |'a'
0x6162636c	1633837903	0x6162634f	'O' |'c' |'b' |'a'
0x61626370	1633837900	0x6162634c	'L' |'c' |'b' |'a'
0x61626374	1633837906	0x61626352	'R' |'c' |'b' |'a'
0x61626378	1633837944	0x61626378	'x' |'c' |'b' |'a'
0x6162637c	1633837927	0x61626367	'g' |'c' |'b' |'a'
