

# CS360 Midterm 1 - January 31, 2019 - 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.
- I have left off header files. Simply assume the correct ones are there.
- The machine is little endian: If an integer is 0xabcd1234, then its 1st byte is 0x34, 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 num);
```

## Question 1

What is the output of the program below?

```
int main()
{
    unsigned int k[4];
    unsigned char *cp;
    unsigned int *ip;
    int i;

    cp = (unsigned char *) k;
    ip = k+2;

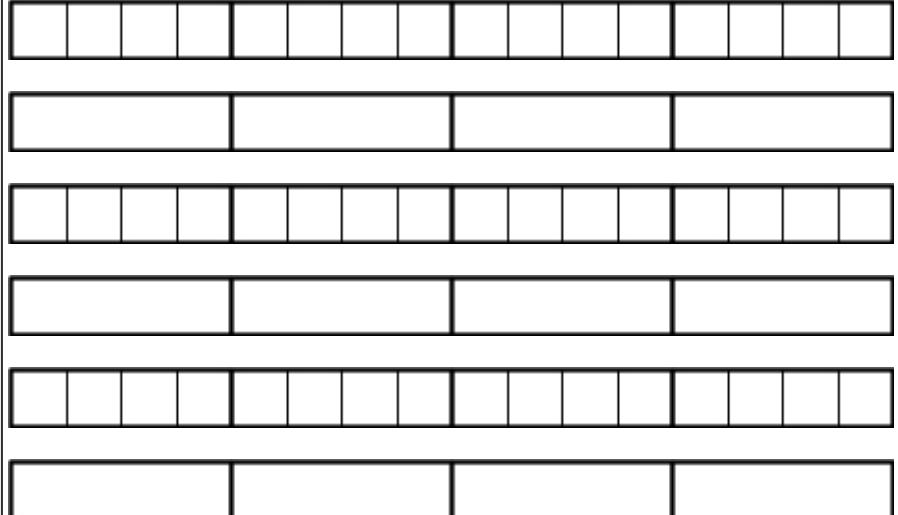
    for (i = 0; i < 16; i++) {
        cp[i] = i*16 + 15-i;
    }

    printf("0: 0x%x\n", cp[3]);
    printf("1: 0x%x\n", cp[7]);
    printf("2: 0x%x\n", k[0]);
    printf("3: 0x%x\n", *ip);

    memcpy(cp+2, cp+10, 4);

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

    cp += 12;
    ip = (unsigned int *) cp;
    i = ip - k;
    printf("8: %d\n", i);
    i = cp - (unsigned char *) k;
    printf("9: %d\n", i);
    return 0;
}
```



# CS360 Midterm 1 - January 31, 2019 - Page 2

## Question 2

What is the output of the program below?

```
int main()
{
    unsigned char a, b, h;
    unsigned int c, d, e, f, g;

    c = 0xc8f66e32;
    d = 0x7f585e9d;
    e = 0xf6543e82;
    f = 0x0ff0f00f;
    a = 0xd9;
    h = 0x4e;
    b = a << 4;
    g = h;
    g <= 4;

    printf("0: 0x%x\n", (c << 12));
    printf("1: 0x%x\n", (d >> 16));
    printf("2: 0x%x\n", e & f);
    printf("3: 0x%x\n", c & ~(f));
    printf("4: 0x%x\n", d | f);
    printf("5: 0x%x\n", (c ^ d) ^ c);
    printf("6: 0x%x\n", (e & 0xffff00) << 8);
    printf("7: 0x%x\n", (a ^ 0xff));
    printf("8: 0x%x\n", b);
    printf("9: 0x%x\n", g);

    return 0;
}
```

## Question 3

What is the output of the program below:

```
int main()
{
    char *x, *y, b[60];
    int i;

    x = b+4;
    y = x+6;

    for (i = 0; i < 59; i++) b[i] = 'A';
    b[59] = '\0';

    strcpy(b, "Smith");
    strcat(b, "Pearson");
    printf("0: %s\n", b);
    printf("1: %s\n", x);
    printf("2: %s\n", y);

    strcpy(b, "Schema-Gallery");
    strcpy(x, "Binky");
    printf("3: %s\n", b);
    printf("4: %s\n", y);

    strcat(y, "Tent");
    strcat(x, "Fly");
    printf("5: %s\n", b);
    printf("6: %s\n", y);

    for (i = 0; i < 50; i += 5) {
        strcpy(b+i, "Friend");
    }
    strcpy(y, "Tater");
    strcpy(b+20, "Tot");

    printf("7: %s\n", x);
    printf("8: %s\n", y+7);
    printf("9: %c\n", x[2]);
    return 0;
}
```

012345678901234567890123456789012345678901234567890123456789

# CS360 Midterm 1 - January 31, 2019 - Page 3

## Question 4

When the following procedure is called, the value of **x** is 0x7652cfe0, and the contents of the 64 bytes starting with **x** are as follows. I'm printing sets of four bytes in hexadecimal and in decimal.

Address	Hexadecimal	Decimal
0x7652cfe0	0x7652cff8	1985138680
0x7652cfe4	0x7652d010	1985138704
0x7652cfe8	0x7652cff4	1985138676
0x7652cfec	0x7652cffc	1985138684
0x7652cff0	0x7652cfe4	1985138660
0x7652cff4	0x7652cfec	1985138668
0x7652cff8	0x7652d008	1985138696
0x7652cffc	0x7652cff8	1985138680
0x7652d000	0x7652d004	1985138692
0x7652d004	0x7652d010	1985138704
0x7652d008	0x7652cff8	1985138680
0x7652d00c	0x7652d000	1985138688
0x7652d010	0x7652cfe4	1985138660
0x7652d014	0x7652d004	1985138692
0x7652d018	0x7652d00c	1985138700
0x7652d01c	0x7652d008	1985138696

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

```
void a(unsigned int *x)
{
    unsigned int **e, ***f;

    e = (unsigned int **) x;
    f = (unsigned int ***) x;

    printf("0: %d\n", *x);
    printf("1: %d\n", x[3]);
    printf("2: %d\n", x[14]);
    printf("3: %d\n", **e);
    printf("4: %d\n", e[4][0]);
    printf("5: 0x%x\n", e[3][2]);
    printf("6: 0x%x\n", ***f);
    printf("7: 0x%x\n", f[2][0][0]);
    printf("8: 0x%x\n", f[1][1][2]);
    printf("9: 0x%x\n", (unsigned int) (e+12));
}
```

## Question 5

When the procedure to the below is called, the value of **a** is `0x41696f40`, and the contents of the 64 bytes starting with **a** are as shown below. Each set of four bytes is printed in decimal, hexadecimal, and as four characters.

Address	Decimal	Hexadecimal	As four characters			
0x41696f40	2780278	0x002a6c76	'v'	'l'	'*'	'\0'
0x41696f44	1097428802	0x41696f42	'B'	'o'	'i'	'A'
0x41696f48	2019819640	0x78640078	'x'	'\0'	'd'	'x'
0x41696f4c	1819869285	0x6c790065	'e'	'\0'	'y'	'l'
0x41696f50	1097428840	0x41696f68	'h'	'o'	'i'	'A'
0x41696f54	1097428804	0x41696f44	'D'	'o'	'i'	'A'
0x41696f58	2037938549	0x79787975	'u'	'y'	'x'	'y'
0x41696f5c	1097428806	0x41696f46	'F'	'o'	'i'	'A'
0x41696f60	7567727	0x0073796f	'o'	'y'	's'	'\0'
0x41696f64	1097428809	0x41696f49	'I'	'o'	'i'	'A'
0x41696f68	1097428806	0x41696f46	'F'	'o'	'i'	'A'
0x41696f6c	1802531437	0x6b70726d	'm'	'r'	'p'	'k'
0x41696f70	28160	0x00006e00	'\0'	'n'	'\0'	'\0'
0x41696f74	1097428848	0x41696f70	'p'	'o'	'i'	'A'
0x41696f78	1870230784	0x6f797500	'\0'	'u'	'y'	'o'
0x41696f7c	1701904502	0x65710076	'v'	'\0'	'q'	'e'

Please tell me the output of the following procedure:

<pre> typedef struct {     int i;     char b;     char *s; } S1;  typedef struct {     int i;     char b; } S2;  typedef struct {     int i;     char *s; } S3; </pre>	<pre> void pm(char *a) {     S1 *x;     S2 *y;     S3 *z;      x = (S1 *) a;     y = (S2 *) a;     z = (S3 *) a;      printf("0: %c\n", x-&gt;b);     printf("1: %d\n", x-&gt;i);     x++;     printf("2: %s\n", x-&gt;s);     printf("3: 0x%x\n", (unsigned int) &amp;(x[1].b));      printf("4: %c\n", y[2].b);     printf("5: %d\n", y[5].i);     printf("6: 0x%x\n", (unsigned int) &amp;(y[1].b));      printf("7: %s\n", z-&gt;s);     printf("8: %d\n", z[2].i);     z += 3;     printf("9: %s\n", z-&gt;s);      return; } </pre>
--	---