Problem 1 (20 points): Write a complete C program that will print the remainder of an integer $x$ when it is divided by an integer $y$, where $-2^{31} \leq x,y \leq 2^{31} -1$. The values of $x$ and $y$ are to be entered through the keyboard. The printed output should read (assuming $x = 12$, and $y = 5$):

The remainder of $x/y$ is: 2

Enter your answer in the box below:
Problem 2 (20 points): Write a complete C program that will print the number of times a character occurs in a string of 50 characters. The character x, and string s are to be entered through the keyboard. The output printed output should read (assuming x = 'a' and s = "banana cake".

The character a occurs 4 times in banana cake.

Enter your answer in the box below:
**Problem 3 (10 points):** What does the function f shown on the left below do?

```c
void f(void)
{
    int c;
    c = getchar();
    while(c != EOF)
    {
        if (c >= 'a' && c <= 'z')
            putchar(c + ('A'-'a'));
        else
            putchar(c);
        c = getchar();
    }
}
```

Enter your answer in the box below and explain it:
Problem 4 (10 points): Write down the output exactly as it will appear on the standard output when the program on the left below is compiled and executed.

```
int main(void)
{
    int a=1, b = 2;
    printf("%d %d\n\n", a, b);

    {int a = 3; int b = 4;
     printf("%d %d\n\n", a, b);
    {int a = 5; int b = 6;
     printf("%d %d\n\n", a, b);
    }
    printf("%d %d\n\n", a, b);

    printf("%d %d\n\n", a, b);

    return 0;
}
```

Enter your answer in the box below and explain it:
Problem 5 (20 points): Write down the output exactly as it will appear on the standard output when the program below is compiled and executed. Specify what each of the functions a, b, and c does.

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

struct member
{
  char *name;
  struct member *next;
};

void b (struct member *head, int x)
{
  struct member *p, *z;
  int i = 0;
  if (head == NULL)
    return;
  for (p = head; (p->next != NULL) && (i < x); p = p->next)
    i++;
  if (p->next == NULL)
    return;
  z = p->next;
  p->next = p->next->next;
  free (z);
}

void c (struct member *head)
{
  struct member *s;
  printf ("The Citizens of Springfield:\n");
  for (s = head; s != NULL; s = s->next)
    printf ("%s \n", s->name);
}

int main (void)
{
  struct member *members = NULL;
  a (&members, "Moe");
  a (&members, "Ned Flanders");
  a (&members, "Homer J. Simpson");
  a (&members, "Montgomery Burns");
  a (&members, "Principal Skinner");
  c (members);
  b (members, 2);
  b (members, 9);
  c (members);
  return 0;
}
void a (struct member **headp, char *name)
{
  struct member *p, *newmemb;
  newmemb = malloc(sizeof(struct member));
  newmemb->name = name;
  if (*headp == NULL)
  {
    newmemb->next = *headp;
    *headp = newmemb;
  }
  else
  {
    for (p = *headp;
      p->next != NULL; p=p->next);
    p->next = newmemb;
    newmemb->next = NULL;
  }
}
```

Enter your answer in the box below.
Problem 6 (20 points): Write down what will be displayed when you open the file, FINAL after the program on the left is compiled and executed assuming that the file was empty before the program was executed.

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

int main(void)
{
    FILE *file;
    char h[6] = "ypahp";
    char hh[9] = "idols";
    char i, j;
    file = fopen("FINAL","w");

    i = h[0];    h[0] = h[3]; h[1] = h[2];

    i = hh[0];     hh[0] = h[0];  j = hh[1];

    for (i=0; i<= 4; i++)
    {
        fprintf(file,h); fprintf(file," ");
        fprintf(file,hh);fprintf(file,"
");
    }
    fclose(file);
    return 0;
}
```

Enter your answer in the box below.

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

int main(void)
{
    FILE *file;
    char h[6] = "ypahp";
    char hh[9] = "idols";
    char i, j;
    file = fopen("FINAL","w");

    i = h[0];    h[0] = h[3]; h[1] = h[2];

    i = hh[0];     hh[0] = h[0];  j = hh[1];

    for (i=0; i<= 4; i++)
    {
        fprintf(file,h); fprintf(file," ");
        fprintf(file,hh);fprintf(file,"
");
    }
    fclose(file);
    return 0;
}
```