Problem 1 (20 points): Write down the output exactly as it will appear on the standard output device when you compile and execute the program shown below:

```
#include <stdio.h>
void y(); void z();
void w(int a); int x(void);
int vector[100]; static *r;

int main(void)
{r = vector;
y(); z(); return 0;}

void w(int a){r++; *r = a;}
int x(void)
{int a = *r; r--; return a;}
void z()
{int i;
for(i=1; i<= 10; i++)
{printf("\n%d\n",x());} }

void y()
{int i;
for(i=1; i<= 10; i++) w(2*i);
}
```

Enter your answer in the box below:
**Problem 2 (20 points):** Write down the binary representations of the C constants that have been declared on the left. Assume that `short` is 16 bits, `int` and `float` are 32 bits each.

Enter your answer in the box below:

```
unsigned char   a = 12;
short           b = -23;
int             c = 25;
float           d = -24.0
```

**Problem 3 (20 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.

Enter your answer in the box below:

```c
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i,j;
    int A[3][3] = {1,2,3,4,5,6,7,8,9};
    int a;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d",A[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<3;i++)
    for(j=0;j<3;j++)
    {
        if(i<j)
        {
            a= -A[i][j]; A[i][j] = -A[j][i];
            A[j][i] = a;
        }
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d",A[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```
Problem 4 (20 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.

```c
#include<stdio.h>
int main(void)
{
    int a=1, b=2, *c, *d, **e, ***f;
    c=&a; d=&b; e=&c; f=&d;

    for(a=0; a<4; ++a)
    {
        switch(a)
        {
            case 0: ***f=**e * 9 * *d/13 +
                    **f + 4 - *d;
            f = &c;
            case 1: *d = 2; b = **f * *d;
                    c = d;
                    break;
            case 2:
                    c = &a; **e = 2 * ***f + 2;
                    ***f = a + *c - 2;
                    *d = 11 -a + b;
            break;
            case 3: *c = 19 + ***f;
            f = &d; a = ***f + *d * *c;
            default: a=6; b=4;
        }
    }

    printf("%d, %d, %d, %d, %d, %d\n",
                    a, b, *c, *d, **e, ***f);

    return 0;
}
```

Enter your answer in the box below:
**Problem 5 (20 points):** Write a C program that reads in a user-specified number of words and prints them out in the reverse order they are entered, with each word written backwards, one per line. So, your program must:

- read in a number n from the user (n must be between 1 and 30)
- read in n words
- print out the words that are read in the reverse order they are entered, with each word's letters written in the reverse of their input ordering

An example run of the program would look like:

Enter the number of words: 6

Enter the words:

Jordan
Kobe
Iverson
McGrady
Webber
Stackhouse

And the words backwards and in reverse order:

esuohkcatS
rebbW
ydargcM
nosrevI
eboK
nadroJ

Your program should be complete and must not return any errors when it is compiled and executed for full credit.