Program "number conversion"
Write a program that converts a number from the decimal system into the binary system by using a recursive function. The converted number should be printed on the screen immediately and should not be stored in any variable (e.g. integer array).

Use the following function call:

```
      void convert_to_binary (int decimal_number);
```

Program "to the power of"
Write a program that calculates the power of a number x^y and use a recursive function inside. Negative values for `x` and `y` are allowed.

Use the following function call:

```
      double power_of (int basis, int exponent);
```

Program "mystrcpy"
Write a program that implements a function called `mystrcpy()` and copies the content of a source array into a target array (incl. end-of-string indicator). This function should work in the same way like `strcpy()`. Do not use this function inside!

Program "leftstr"
Write a program that implements the function `leftstr(char * target, const char * source, size_t n)`. This function should copy only `n` characters from the source to the target location. The target has to end with the end-of-string indicator. Please ensure that the function `leftstr()` does not copy more elements from the source than available in the target location. Do not use any function from `sting.h`.

Program "reverstr"
Write a program that writes a string backwards on the screen. The users should input a word or sentence and the program immediately presents the output. First implement this without a sub-function (only main function) then try to program a sub-function that does the backward-writing and another function that does the screen output.

Program "upstr"
Write a program that converts all characters from an array into capital letters. Please ensure that letters that are already capital letters stay unchanged and any special characters are not converted into other curios characters.

Program "password"
Write a program that reads a password from the keyboard but does not show this on the screen. As a second step the program has to check whether the password follows some specific rules (e.g. password contains more than 8 characters). After the check the user should decide whether the program shows the password on the screen or not.

Program “guess a random number”

The header file stdlib.h contains a function to generate a random number: rand(). The value that you can get from rand() are between 0 and 2^{15}-1. If the function rand () is called the random number are calculated with a specific algorithm. The downside of that algorithm is that the result is always the same after each function call. To change that behaviour you need to initialize the start value you can use the function srand() which initializes the random generator with its input variable:

```
int startvalue;
printf (“please enter a start value: “);
scanf(“%i”, &startvalue);
srand (startvalue);
printf (“the random number is: %i\n”, rand());
```

If you want to create a real random number you can use the function time() described in the header file time.h. This function takes a variable of the data type time_t which is also describes in time.h and returns the number of seconds between now and January 1st, 1970. Under the assumption that you will not call the function time exactly after 1 second you can use time() to initialize the random generator.

1. Write a program that uses that mechanism and initializes the random generator with the time() function. The program’s output should be a random number in the range of values between min and max.

2. With the extension of the program the user should be able to guess the random number and once the user has done the right guess the program tells the user the also the number of attempts.

Example:

```
Range of random numbers is 20…30.
Your guess is:
1. try: 23        wrong guess
2. try: 24        wrong guess
3. try: 21        right!
You have got the number within 3 attempts.
```

3. Write another extension that makes it possible to play that game with 2 players which have to guess the number within a pre-defined range of values e.g. 1…100. Tell the player whether the number is greater or less than the random number.

The winner gets a coffee from the loser. ☺