Problem 1

Write C function which takes 16 bit number and should determine whether exactly one bit in this number is set. If one bit is set, the function should return "1" otherwise "0". The prototype of function is "unsigned char CheckBit(unsigned int uValue)".

Example:
Input = 0x0400, Output = 0x01
Input = 0x0600, Output = 0x00
Input = 0x0000, Output = 0x00

Problem 2

Realize function that reverses null terminated string. The prototype of the function is "void Reverse(char* ptr);". Do not use standard library functions.

Example: If e.g. “This is a test\0” is passed to the function, the function must produce “tset a si sihT\0”

Problem 3

Write a C function which takes as input 8x8 pixels image and returns a vertical mirror of the image. Image is monochrome - each pixel is coded as one bit and each byte represents one image row.

The prototype of the function is: "void VMirror(unsigned char *pImage)".

Example:

initial content of buffer : 0x7F,0x40,0x40,0x40,0x7F,0x00,0x00,0x00

after VMirror execution : 0xFE,0x02,0x02,0x02,0xFE,0x00,0x00,0x00

Problem 4

Write C function which takes signed 16 bit number (int) and convert it into ASCII null terminated string. Function prototype is "void Conv(int sValue, char *pBuffer)", where pBuffer is pointer to string buffer. Format of the string should be "-DDDDD", if the number is positive the first character is space. The string should not contain leading zeroes.

Example:

```
"-   26"
" 16025"
```

Problem 5

Realize queue from integers with array (ring buffer). Push and Pop function should be realized, e.g.:

```
int arr[ MAX_SIZE ];
int top = 0;
int bottom = 0;
```

int IsEmpty() - functions returns 0 if queue is empty and > 0 if not

int IsFull() - function return 0 if the queue is full and new element cannot be added, otherwise > 0

void Push(int nNewValue) - Functions adds new element to the queue. We assume, that previously "IsFull()" was called to check whether is possible to insert element, i.e. it is always possible to insert element

int Pop() - retrieve an element from the queue. We assume that the queue is not empty.

Problem 6

Write a function, which performs bubble sort. The prototype of the function is "void Bubble(int* pArray, unsigned int uLen)", where pArary is a pointer to unsorted array and uLen is its size.

Problem 7

Write a function which performs binary search. The prototype of the function is:

"unsigned int BinSearch(unsigned int *pArray, unsigned int uArraySize, unsigned int uValue)", where pArray is pointer to sorted array with ascending order of elements, uArraySize is number of elements in the array and uValue is the searched value. If the uValue is found, the function should return it index, otherwise 0xFFFF.

Problem 8

Write a function, which performs addition of positive numbers, represented as strings. The result is stored into a buffer, passed as parameter to the function. We assume that the result buffer is always big enough. The prototype of the function is "void Add(const char* p1, const char* p2, char* result);"

Example:

if the function is called as follows: Add(“12345”, “678”, result) , the content of “result” must be “13023” or " 13023".

Problem 9

Write simple “sprintf” function. The prototype of the function is "void SPRINTF(char* szBuffer, char* szPattern, char** args);” The “szBuffer” is the resulting string, “szPattern” is the format control. If and only if “%s” appears in the format string “szPattern”, it is replaced with the corresponding next argument from the “args”.

Example:
const char* args[ ] = { “Arg1”, “Arg2” };

SPRINTF(szBuffer, “This function takes %s and %s as arguments\n”, args); ,

Should print: This function takes Arg1 and Arg2 as arguments\n”

Problem 10

Write a declaration of class cWindow with following integer member variables (properties) : Left, Right, Top, Bottom, which are initialized by the default constructor. The class should have a copy constructor. Write a declaration of derived class cHint with additional member variable sCaption of type tString. Write a declaration of another derived class cBitButton with additional member variable aBitmap of type tBitmap. Both derived classes should have a default constructor which initializes all member variables.