Q1. What will be the output of this code? Provide a brief explanation.

```
int main() {
  char ca = 'a';
  char cb = 'b';

  char* const ptr1_char = &ca;
  char* ptr2_char = &cb;
  ptr2_char = ptr1_char;

  printf("%c", *ptr1_char);
  printf("%c", *ptr2_char);
  return 0;
}
```

Q2: Explain what's wrong with this code. Compile it in CSIL to see if it compiles and what it outputs. Compile it in replit to see if it compiles and what it outputs.

```
#include <stdio.h>

int main() {
  const char c = 'a';

  char* ptr_char = &c;
  *ptr_char = 'B';

  printf("%c", c);
  return 0;
}
```

Q3: Explain what's wrong with this code. Compile it in CSIL to see if it compiles and what it outputs. Compile it in replit to see if it compiles and what it outputs.

```
#include <stdio.h>

int main() {
  char ca = 'a';
  char cb = 'b';

  char* const ptr1_char = &ca;
  char* ptr2_char = &cb;
  ptr2_char = ptr1_char;
  ptr1_char = ptr2_char;

  printf("%c", *ptr1_char);
  printf("%c", *ptr2_char);
  return 0;
}
```

Q4: Write a function that gets an array of ints of length n, and returns the maximum value.

```
int max(const int array[], int n)
```

Q5: Write a function that gets an array of ints of length n, and returns the sum of the two largest values.

```
int sum_max2(const int array[], int n)
```

Q6: Write a function that gets a 2d array of ints, and checks if it contains two equal rows.

```
bool contains_equal_rows(int height, int width,
                         const int array[height][width])
```

Q7: Write a function that gets an array of digits (given as ints) of length n, and prints the largest number possible using these digits.

```
void print_max_number(const int digits[], int n)
```

For example: on input {7, 5, 3, 8, 3, 0} the function should print 875330.

*Note: you are not required to return the number because it may be too large.

Q8: Write a function that gets two arrays of ints a and b, and checks a[i] < b[i] for all i.

Decide on the correct signature for the function. Implement it.

Q9: Write a function that gets a string (array of chars ending with '\0') and reverses it in place.

```
void reverse(char str[])
```

For example, on input "hello", str should become "olleh".

Q10: Write a function that gets a string str and a char c, and returns the index of the first appearance of c in str. If c is not in str, the function returns -1.

```
int str_find(const char* str, char c)
```

For example, on input str="ABCa!i!aaD" and c='a' the function returns 3. on input str="ADaCFaDDa" and c='b' the function returns -1.

Q11: Write a function that gets a string (array of chars ending with '\0') and checks if it is a palindrome.

```
bool is_palindrome(const char* str)
```

For example, it outputs true if the input is “racecar”, but returns false if the input is “engage”.

Q12: Write a function that gets a string (array of chars ending with '\0') and returns the longest substring that is palindrome. The returned string should be allocated on the heap.

```
char* longest_substring_palindrome(const char* str)
```

For example, on the input is "ABCBDEFFFEDS", it needs to return "DEFFFED". Solve it in O(n^2) time.

Q13: Write a function that gets an array of ints of length n, and outputs the length of the longest contiguous increasing subsequence.

```
int longest_incr_subsequence(const int arr[], int n)
```

For example, on input {1, 2, 6, 4, 3, 8, 9, 10, 2, 4} the output should be 4 because {3, 8, 9, 10} is the longest subsequence.

Q14: Write a function that gets a string that contains a number in binary, and returns the number as int. Assume that the number can fit into an int.

```
int binary_to_int(const char* binary)
```

For example, on input "11011" the return is 27.

Q15: Write a function that gets a string that contains a number in binary, and returns a string containing the number in decimal. You should not assume that the number can fit into an int or long. You should handle inputs up to length 1,000,000.

```
char* str_binary_to_dec(const char* binary)
```

For example, on input "11011" the return is "27".

Q16: We said that int take 4 bytes in the memory. Write a program that checks the order in which the number is stored in the memory. Does it start from most significant bits or from least significant bit? (most significant bit refers to the bit that stores the largest value. E.g., if “10” means 2, it’s the left bit.)