1.

```c
#include <stdio.h>

#define MAX 1000

int main() {
    int nums[MAX], target, n, i, j;

    // Read the number of elements and the target
    scanf("%d", &n);
    scanf("%d", &target);

    // Read the array elements
    for(i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }

    // Find the two indices
    for(i = 0; i < n; i++) {
        for(j = i + 1; j < n; j++) {
            if(nums[i] + nums[j] == target) {
                printf("%d %d\n", i, j);
                return 0;
            }
        }
    }

    return 0;
}
```

```
/tmp/IUDINQU8xX.o
5
3
0 1 2 4 6
1 2


=== Code Execution Successful ===
```

2.

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

// Definition of the linked list node
typedef struct ListNode {
    int val;
    struct ListNode *next;
} ListNode;

// Function to create a new node
ListNode* createNode(int value) {
    ListNode* newNode = (ListNode*)malloc(sizeof(ListNode));
    newNode->val = value;
    newNode->next = NULL;
    return newNode;
}

// Function to add two numbers represented by linked lists
ListNode* addTwoNumbers(ListNode* l1, ListNode* l2) {
    ListNode *dummyHead = createNode(0);
    ListNode *current = dummyHead;
    int carry = 0;

    while (l1 || l2 || carry) {
        int sum = carry;
        if (l1) {
            sum += l1->val;
            l1 = l1->next;
        }
```

```
/tmp/BVj3HzH7KT.o
7 0 8


=== Code Execution Successful ===
```

```
        if (l2) {
            sum += l2->val;
            l2 = l2->next;
        }
        carry = sum / 10;
        current->next = createNode(sum % 10);
        current = current->next;
    }

    return dummyHead->next;
}
void printList(ListNode* head) {
    while (head) {
        printf("%d ", head->val);
        head = head->next;
    }
    printf("\n");
}
int main() {
    ListNode* l1 = createNode(2);
    l1->next = createNode(4);
    l1->next->next = createNode(3);
    ListNode* l2 = createNode(5);
    l2->next = createNode(6);
    l2->next->next = createNode(4);
    ListNode* result = addTwoNumbers(l1, l2);
    printList(result);
    return 0;
}
```

3.

```
#include <stdio.h>
#include <string.h>
#define MAX_CHARS 128

int lengthOfLongestSubstring(char* s) {
    int lastIndex[MAX_CHARS] = {-1};  // Store the last index of each character
    int maxLength = 0, start = 0;

    for (int i = 0; s[i]; i++) {
        char c = s[i];
        if (lastIndex[c] >= start) {
            start = lastIndex[c] + 1;
        }
        lastIndex[c] = i;
        if (i - start + 1 > maxLength) {
            maxLength = i - start + 1;
        }
    }

    return maxLength;
}

int main() {
    char s[] = "abcabcbb";  // Example input
    printf("%d\n", lengthOfLongestSubstring(s));  // Output: 3
    return 0;
}
```

```
/tmp/WTdOrPI2SV.o
3


=== Code Execution Successful ===
```

4.

```python
def findMedianSortedArrays(nums1, nums2):
    if len(nums1) > len(nums2):
        nums1, nums2 = nums2, nums1
    m, n = len(nums1), len(nums2)
    imin, imax, half_len = 0, m, (m + n + 1) // 2

    while imin <= imax:
        i = (imin + imax) // 2
        j = half_len - i
        if i < m and nums1[i] < nums2[j - 1]:
            imin = i + 1
        elif i > 0 and nums1[i - 1] > nums2[j]:
            imax = i - 1
        else:
            max_left = max(nums1[i - 1] if i > 0 else float('-inf'),
                           nums2[j - 1] if j > 0 else float('-inf'))
            if (m + n) % 2 == 1:
                return max_left
            min_right = min(nums1[i] if i < m else float('inf'),
                            nums2[j] if j < n else float('inf'))
            return (max_left + min_right) / 2.0

if __name__ == "__main__":
    print(findMedianSortedArrays([1, 3], [2]))  # Output: 2.0
    print(findMedianSortedArrays([1, 2], [3, 4]))  # Output: 2.5
```

```
2
2.5

=== Code Execution Successful ===
```

5.

```c
#include <stdio.h>
#include <string.h>
int main() {
    char s[] = "babad";  // Example input
    int len = strlen(s);
    int start = 0, max_len = 1;
    for (int i = 0; i < len; i++) {
        int l = i, r = i;
        while (l >= 0 && r < len && s[l] == s[r]) {
            if (r - l + 1 > max_len) {
                start = l;
                max_len = r - l + 1;
            }
            l--;
            r++;
        }
        l = i, r = i + 1;
        while (l >= 0 && r < len && s[l] == s[r]) {
            if (r - l + 1 > max_len) {
                start = l;
                max_len = r - l + 1;
            }
            l--;
            r++;
        }
    }
    printf("%.*s\n", max_len, s + start);
    return 0;
```

```
/tmp/rWSxErUnOh.o
bab


=== Code Execution Successful ===
```

7.

```python
def reverse(x):
    INT_MIN, INT_MAX = -2**31, 2**31 - 1
    sign = -1 if x < 0 else 1
    x = abs(x)

    reversed_num = 0
    while x != 0:
        digit = x % 10
        x //= 10
        if reversed_num > (INT_MAX - digit) // 10:
            return 0

        reversed_num = reversed_num * 10 + digit

    return sign * reversed_num

# Example usage
print(reverse(123))
print(reverse(-123))
print(reverse(120))
```

```
321
-321
21

=== Code Execution Successful ===
```

8.

```python
def myAtoi(s):
    INT_MIN, INT_MAX = -2**31, 2**31 - 1
    i, n = 0, len(s)
    while i < n and s[i] == ' ':
        i += 1
    if i < n and s[i] in '+-':
        sign = -1 if s[i] == '-' else 1
        i += 1
    else:
        sign = 1
    result = 0
    while i < n and s[i].isdigit():
        digit = int(s[i])
        if result > (INT_MAX - digit) // 10:
            return INT_MAX if sign == 1 else INT_MIN
        result = result * 10 + digit
        i += 1
    result *= sign
    return max(INT_MIN, min(result, INT_MAX))
print(myAtoi("42"))
print(myAtoi("   -42"))
print(myAtoi("4193 with words"))
print(myAtoi("words and 987"))
print(myAtoi("-91283472332"))


```

```
42
-42
4193
0
-2147483648

=== Code Execution Successful ===
```

9.

```python
def isPalindrome(x):
    if x < 0 or (x % 10 == 0 and x != 0):
        return False
        original, reversed_num = x, 0
    while x > 0:
        reversed_num = reversed_num * 10 + x % 10
        x //= 10

    return original == reversed_num
print(isPalindrome(121))
print(isPalindrome(-121))
print(isPalindrome(10))

```

```
True
False
False

=== Code Execution Successful ===
```

10.

```python
def isMatch(s, p):
    dp = [[False] * (len(p) + 1) for _ in range(len(s) + 1)]
    dp[0][0] = True
    for j in range(1, len(p) + 1):
        if p[j - 1] == '*':
            dp[0][j] = dp[0][j - 2]
        for i in range(1, len(s) + 1):
            for j in range(1, len(p) + 1):
                if p[j - 1] == s[i - 1] or p[j - 1] == '.':
                    dp[i][j] = dp[i - 1][j - 1]
                elif p[j - 1] == '*':
                    dp[i][j] = dp[i][j - 2] or (dp[i - 1][j] and (p[j - 2] == s[i - 1] or p[j
                        - 2] == '.'))

    return dp[len(s)][len(p)]

# Example usage
print(isMatch("aa", "a"))
print(isMatch("aa", "a*"))
print(isMatch("ab", ".*"))
print(isMatch("aab", "c*a*b"))
print(isMatch("mississippi", "mis*is*p*."))


```

```
False
True
True
True
False

=== Code Execution Successful ===
```