Assignment 1 — Part 2

Task 1: Reverse Numbers

Write a function that takes in an integer ‘num’ and returns the reverse integer number:

Example 1:

```
Input:

num: 1234
```

```
Output:

4321
```

Example 2:

```
Input:

num: 784592749214
```

```
Output:

412947295487
```

Task 2: Differences between lists of words

Write a function that takes two input variables (lista and listb) which you can safely assume to be lists containing multiple values which can only be strings with English letters (no commas, full stops or similar).

For both of these lists, let the function print out the following on individual lines:

1. The intersection of both sets

2. Which items are in list A but not in list B

3. All items of both lists, but without any duplicates

Example 1:

Input:

```
lista = ['the', 'new', 'york', 'times', 'is', 'an', 'american', 'newspaper', 'based', 'in', 'new', 'york', 'city']

listb = ['the', 'neue', 'zürcher', 'zeitung', 'is', 'a', 'swiss', 'german-language', 'daily', 'newspaper']
```

Output:

```
Intersection: {'the', 'newspaper', 'is'}

Difference: {'new', 'in', 'city', 'york', 'american', 'based', 'an', 'times'}

Union: {'in', 'york', 'swiss', 'american', 'based', 'an', 'newspaper', 'german-language', 'new', 'zürcher', 'daily', 'city', 'neue', 'is', 'a', 'the', 'times', 'zeitung'}
```

Task 3: The longest substring

Given a string `s`, find the length of the **longest substring** without repeating characters.

Example 1:

```
Input: s = "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.
```

Example 2:

```
Input: s = "bbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.
```

Example 3:

```
Input: s = "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3.

Notice that the answer must be a substring, "pwke" is a subsequence and not a
substring.
```

Example 4:

```
Input: s = ""

Output: 0
```

Constraints:

- `The length of the string is >0 and <= 100 characters`
- `The string only consists of English letters`

Extra Task: Kids with Candy

Given the list `candies` and the integer `extra_candies`, where `candies[i]` represents the number of candies that the ***ith*** kid has.

For each kid check if there is a way to distribute `extraCandies` among the kids such that he or she can have the **greatest** number of candies among them. Notice that multiple kids can have the **greatest** number of candies.

Example 1:

```
Input: candies = [2,3,5,1,3], extra_candies = 3

Output: [true,true,true,false,true]

Explanation:

Kid 1 has 2 candies and if he or she receives all extra candies (3) will have 5
candies --- the greatest number of candies among the kids.

Kid 2 has 3 candies and if he or she receives at least 2 extra candies will have
the greatest number of candies among the kids.

Kid 3 has 5 candies and this is already the greatest number of candies among the
kids.

Kid 4 has 1 candy and even if he or she receives all extra candies will only have
4 candies.

Kid 5 has 3 candies and if he or she receives at least 2 extra candies will have
the greatest number of candies among the kids.
```

Example 2:

```
Input: candies = [4,2,1,1,2], extra_candies = 1

Output: [true,false,false,false,false]

Explanation: There is only 1 extra candy, therefore only kid 1 will have the
greatest number of candies among the kids regardless of who takes the extra candy.
```

Example 3:

```
Input: candies = [12,1,12], extra_candies = 10

Output: [true,false,true]
```

Constraints:

- `2 <= candies.length <= 100`
- `1 <= candies[i] <= 100`
- `1 <= extraCandies <= 50`