SU
SINAI UNIVERSITY
Arish - Kantara

Faculty of Information Technology and Computer Science

Computer Programming 1 – Sheet Single Loop – Playing with digits

Single Loops

1) Write a C++ program to read an integer x and reverse it.
2) Write a C++ program to read an integer x and check if the number is palindrome or not. (العكس هو الأصل)
    - If x=717 then it is palindrome
    - If x=231 then it is not palindrome because when reversed → 132
    - If x=81218 then it is palindrome because when reversed → 81218
3) Write a C++ program to read an integer and compute the sum of the squares of its digits.
4) Write a C++ program that reads two integers x and y then check if the sum of digits in x is greater, less or equals to the sum of digits in y.
5) Write a C++ program that reads two integers x and y then check if the sum of the first half digits in x is greater, less or equals to the sum of the second half digits in y.
    - If x=4316, y=12 compare 4+3 and 2
    - If x=43165, y=1256 compare 4+3 and 5+6
6) An Armstrong number is a special kind of number in math. It's a number that equals the sum of its digits, each raised to a power n where n is the number of digits. For example, if you have a number like 153, it's an Armstrong number because 1^3 + 5^3 + 3^3 equals 153. 1634 is an Armstrong number because 1^4+6^4+3^4+4^4=1634. Write a C++ program to read an integer x and check whether it is Armstrong or not.
7) Erimp is a prime number that results in a different prime when its decimal digits are reversed (prime before and after reversal). Write a C++ program to read an integer x and check whether it is Erimp or not.
    - If x=17 then it is Erimp because 17 is prime and when reversed it gives us 71 which is also a prime number
    - If x=15 then it is not Erimp because 15 is not a prime number
    - If x=113 then it is Erimp because 113 is prime and when reversed it gives us 311 which is also a prime number

    Here is the list of Erimp numbers: https://prime-numbers.info/list/emirps
8) An automorphic number is a number whose square ends with the same digits as the original number. Write a C++ program to read an integer x and check if it is automorphic or not.
    - If x=25 then it is automorphic because 25 * 25 = 625, and the last two digits (25) are the same as the original number.
    - If x=76 then it is not automorphic because 76 * 76 = 5776, and the last two digits (76) are different from the original number.
    - If x=376 then it is automorphic because 376 * 376 = 140676, and the last two digits (76) are the same as the original number.

9) Write a C++ program that reads integer x that represents a binary number then convert it to **decimal** and print the result.
    - If x=1101101 output: 109
    - If x=101110   output: 46

10) Write a C++ program that reads integer x that represents a binary number then convert it to **octal** and print the result.
    - If x=1101101 output: 155
    - If x=101110   output: 56

11) Write a C++ ***bitwise calculator*** program that reads two integers x and y both represents binary numbers y then read an operation {&,|} if the user chooses ’&’ perform the **and logical operation** between x and y but if the user chooses ’|’ perform the **or logical operation** between x and y then print the binary result to the user in any of the mentioned scenarios.
    - If x=1011   y=101   op=’&’   result=1
    - If x=1011   y=10010   op=’&’   result=10
    - If x=1011   y=101   op=’|’   result=1111
    - If x=1011   y=10010   op=’|’   result=11011

12) The Hamming distance between two binary numbers of equal length is the number of positions at which the corresponding bits are different. It's used in various applications like error correction and code design. Write a C++ program to read two integer x and y then compute the hamming distance between them.

    Examples:

    If x = 1011 and y = 1001, then the Hamming distance is 1 because only the third bit (from the right) differs.

    If x = 1100 and y = 0110, then the Hamming distance is 3 because the first, second, and fourth bits differ.

    If x = 0000 and y = 1111, then the Hamming distance is 4 because all four bits differ.

13) Write a C++ program to read an integer x, then count the frequency of the number n in x

    Example: If x= 3783433 and n=3   then the output will be = 4
    Example: If x= 5825239 and n=4   then the output will be = 0
    Example: If x= 5125251 and n=51 then the output will be = 2

14) Write a C++ program to read an integer x, then check if the first 3 digits of x equal to the last 3 digits of x. If they are equal print Yes otherwise print No.

    Example: If x = 901465901 then the output will be → Yes
    Example: If x = 4025402 then the output will be → Yes
    Example: If x = 4025400 then the output will be → No

15) Write a C++ program to read two integers x and y, then swap the last digit of x with the last digit of y

    Example: If x = 234 and y = 789 then the output will be x = 734 y = 289

16) Write a C++ program to read an integer x and generate a new number y that contains the second half of x digits in a reverse order.

Example: If x = 4567 then the output will be y=76

Example: If x = 56782 then the output will be y=28

Example: If x = 5678214 then the output will be y=412

17) Write a C++ program to read two integers **x** and **k** where k is the number of digits to be truncated from x and reversed then attached back to x.

If x=45612 k=3 → x= 45216

If x=45612 k=2 → x= 45621

If x=9125 k=1 → x=9125

18) Write a C++ program to read two integers **x** and **y** then insert **y** in the middle of **x**.

If x=4561 y=3 → x= 45361

If x=45618 y=97 → x= 459718

If x=9125 y=7 → x=91725

19) Write a C++ program to read an integer x, then replace all of the first half digits with the max digit and replace all of the second half digits with the min digit.

If x = 345654 output: x= 666333

If x = 32418 output: x= 88811

20) Write a C++ program that reads two integers x and k where k is the number of digits to be rotated in x to the right.

If x = 368792 k=1 output: x= 687923

If x = 368792 k=2 output: x= 879236

If x = 368792 k=4 output: x= 923687

21) Write a C++ program that reads two integers x and k then digit zero will be inserted between every k digits in x.

If x = 3687 k=1 output: x= 3060807

If x = 458796 k=2 output: x= 45087096

If x = 3687921 k=3 output: x= 368079201

If x = 3687921 k=4 output: x= 3687092

22) Write a C++ program that reads two integers x and k then construct a new number y where the sum of each k digits in x forms a new number in y.

If x = 12453 k=1 output: y= 12453 because (1) (2) (4) (5) (3)

If x = 12453 k=2 output: y= 393 because (1+2) (4+5) (3+0)

If x = 12453132 k=3 output: y= 795 because (1+2+4) (5+3+1) (3+2+0)

If x = 12453132 k=4 output: y= 126 because (1+2+4+5) (1+3+2+0)

23) Write a C++ program that reads an integer x and construct a new number y which duplicates each digit in x by its value.

If x = 341 output: x= 33344441

If x = 1302 output: x= 133322

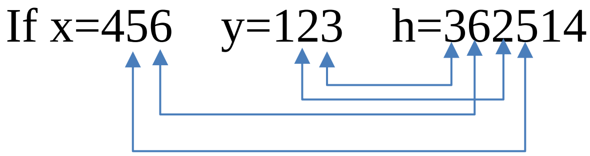
If x = 10213 output: x= 1221333

24) Write a C++ program that reads an integer x then do the following:

- Count the odd digits in x.
- Count the even digits in x.
- Construct a new number y that consists of only the even digits of x.
- Construct a new number v that consists of only the odd digits of x.

Example: If x=567891 Output should be: #odd digits=4 #even digits=2 y=68 v=5791
(order of the digits is important in the output)

25) Write a program to read two integers **x** and **y** then construct a new number **h** that takes alternating digits from x and y. ***(consider that x and y have the same number of digits)***



If x=456   y=123   h=362514

If x=912 y=374   h=427139

26) Write a program to read two integers **x** and **y** then construct a new number **h** that takes alternating digits from x and y.

If x=45629   y=183   h=39821654
If x=183   y=45629   h=93286154