Answers to Part 1, by Roman:

1)Reference is a way to create a second name for a variable.
This allows to modify the value of the original(!) variable without having direct access to it

How to create reference:
(variable type[ex: int, float, bool])& (variable name) = (original variable name);

Example: int& newX = x;

2)
1]If you want to change the value of variable then using "pass by reference" allows to do it much quicker than regular way(pass by value, change the value in function, return the value)
2]If you pass variable by reference then it's value doesn't need to be copied so it's faster/more memory effective

3)
1]You don't need to check to be sure that reference isn't pointing to NULL(unless you're making a reference to zero but that's a special case)
2]You won't meet the problem of uninitialized reference/reference that you didn't allocate memory for

4)
A)Yes
B)Yes
C)No, because you're sending a constant instead of variable, so either the compiler will go to the first memory where he meets this constant, or it'll go to cell number 6 in memory

5)Same problem so I'll say the answer one time:
In given examples you try to give reference to local variables, this variables will be erased the moment you end the function so their reference will be invalid