**Section A: Theory**

1. Every variable takes space in the program. By declaring that variable we actually declare the space that variable takes in order for the program to use the variable properly ; e.g “float 10.5” or “int 10” , will be treated differently because if 10.5 was declared as “int” , the “0.5” wouldn’t appear at all. Also appears an “error”.
2. A “Vector” compared to an “array” , can be easily modified as it comes to size and context. By using simple commands such as “vector\_name.push\_back(context)” we add an extra number to the end of the vector. After that we can identify the new size with another command , “vector\_name.size” giving us the size vector has at the moment. On the other hand arrays cannot do that and remain the same through the whole process.
3. .

* Push\_back: by using “vector\_name.push\_back(5)” we add an extra number at the end of our vector , in that occasion , number 5 .
* Pop\_back : by using “vector\_name.pop\_back” we remove a number from the end of our vector , thus removing size by -1 .
* Size: by using “vector\_name.size” we can see the vector’s size, even after changes we might have done like adding or removing a number.
* Empty: “vector\_name.empty” is a Boolean statement which returns true if vector contains no text and the space it occupies ( e.g. if it has size= 5 by we haven’t ginen any numbers to fill in)
* Clear: “vector\_name.clear” as the name says, it removes any given number or context from vector , leaving it with empty space

**Section B : Understanding**

1. .

&

&

&

&

&

&

&

&

&

&

1. .

While (int i=100 & i>0 & i++)

Cout<<i<<endl;

1. Its illegal because its void and voids do not need “return” statement .
2. Its print the number “10” for one reason. The call we did for the function is incorrect and it should be:

Std::cout<<proc(num)<<std::endl;

1. .
2. Refers to the context of the 2nd position in the vector since 0 is the 1st . So it is “3”.
3. Refers to the size of the vector . size= 5.
4. False
5. 2
6. 14
7. 4