ENSF 614 – Fall 2021
Quiz 1

Some Solutions

Question 1 (7 marks)

Consider the following function prototype in C, and write the definition/implementation of the function in the box below.

```
char* reverse_strcat(char* dest, char* source);
/* REQUIRES: dest and source each one point to a valid C string.
 * PROMISES: all the characters in source will be appended to the end of dest,
 * in reverse order.
 * Then similar to most of the string library functions, it should return the
result.
 * EXAMPLES: Assuming dest points to "ABC" and source points to "WXYZ", the function
 * should returns the result which is "ABCZYXW".
 * Notice that "WXYZ" is appended in reverse order.
 */
```

Important Notes:

In this function, you can **NOT** use any C library function, and your are allowed only to use two local variables in your function as is shown below:

```
char* p;
int i;
```

Marks will not be awarded to a function that uses any local variable other than `char* p`, and `int i`. Also, you can assume that **`dest`** is large enough to hold additional characters from **`source`** points to.

```
char* reverse_strcat(char* dest, char* source) {
    char *p;
    int i;
    i = 0;
    p = dest;

    while(*dest) dest++;

    while(source[i] !='\0') i++;

    while(i > 0) {
        *dest = source[i-1];
        dest++;
        i--;
    }
    *dest = '\0';
    return p;

}
```

Question 2 (8 marks)

Consider the partial definition of the following class called Vector. In the following space, draw an AR diagram for point one, inside function push_back.

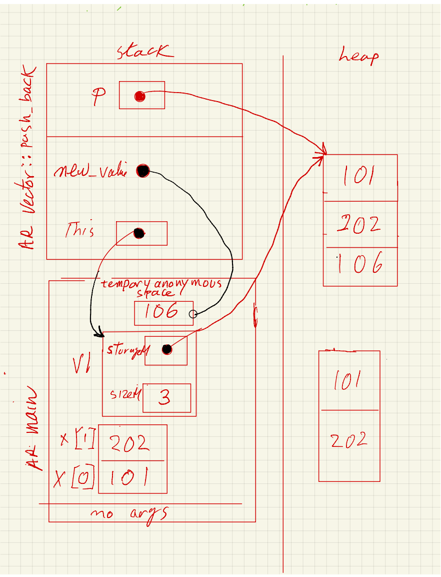
You can assume all necessary header files are included.

```
class Vector {
public:
    Vector();
    Vector(const int *a, int s);
    void push_back(const int& new_value);
    int at(int index) const;
    void set_element(int index, int new_value);
    void display();
private:
    int sizeM;
    int *storageM;
};
//--------------------------------
Vector::Vector(): sizeM(0), storageM(nullptr){
}
Vector::Vector(const int *a, int s): sizeM(s){
    storageM = new int[sizeM];
    assert(storageM != nullptr);
    for (int i = 0; i < sizeM; i++)
    storageM[i] = a[i];
}
void Vector::push_back(const int& new_value)
{
    int *p = new int[sizeM + 1];
    assert(p!=nullptr);
    for(int i=0; i < sizeM; i++)
    p[i] = storageM[i];
    p[sizeM] = new_value;
    sizeM++;
    delete[]storageM;
    storageM = p;

    // POINT ONE

}

int main()
{
    int x[] = { 101, 202 };
    Vector v1(x, 2);
    v1.push_back(106);
    v1.display();
    return 0;
}
```



stack
heap
AR Vector::push_back
p
new_value
This
101
202
106
AR main
tempory anonymous space
106
v1
storageM
sizeM
3
x[1]
202
x[0]
101
no args
101
202