POINTERS EXERCISES

1. What advantages do pointers give us?
2. What operator allocates memory dynamically?
3. What does it really mean to allocate memory? Does it have a name?
4. Why is it important to subsequently deallocate that memory?
5. What operator deallocates memory?
6. What is the disadvantage to a dynamically allocated array?
7. Write a C++ function to demonstrate how a function returns a pointer.
8. Write a C++ code to demonstrate the use of pointer to structures.
9. Write a C++ code to show a pointer to an array whose contents are pointers to structures.
10. Write the code to declare an array of 10 pointers to array of 100 characters.

11. Write the function Extend1Element to extend the size of array to n+1 and copy all elements to this new array. Which one is correct?

```
a. void Extend1Element(int arr1[], int n1, int* arr2, int& n2);
b. void Extend1Element(int arr1[], int n1, int*& arr2, int& n2);
```

12. Implement the following:
    a. Declare an array of five integers and initialize it to the first five odd positive integers.
    b. Write a statement that assigns the sum of the first and last elements of the array in a. to the variable `even`.
    c. Declare an array of chars and initialize it to the string `"cheeseburger"`

13. Implement the following:
    a. Devise a structure declaration that describes a ***fish***. The structure should include the ***kind***, the ***weight*** in whole kilogram, and the length in fractional meters.
    b. Declare a variable of the type defined in a. and initialize it.
    c. Write a code fragment that dynamically allocates a structure of the type described in a. and then reads a value for the ***kind*** member of the structure.

14. Write a C++ function to add two numbers using pointers.

```
int Add(int* a, int* b);
```

15. Write a C++ function to swap two integers using pointers.

```
void Swap(int* a, int* b);
```

16. Write a C++ function that takes in an array of integers and its length, and returns the sum of all the elements.

```
int SumArray(int* a, int len);
```

17. Write a C++ function to find the factorial of a given number using pointers.

```
int Factorial(int* a);
```

18. Write a C++ function to find the length of a string using pointer.

```
int StringLen(char* str);
```


Lecturer: Nguyễn Hải Minh – University of Science, HCM city – nhminh@fit.hcmus.edu.vn

19. Write a function that takes in a string and reverses it using pointers.

```
void Reverse(char* str);
```

20. Write a C++ function to count the number of vowels and consonants in a string using a pointer.

```
void CountVC(char* str, int& vowels, int& consonants);
```

21. Write a function that takes in an integer n and a pointer to an array of n integers and returns a new array that contains the elements of the original array in reverse order.

```
int* ReverseArray(int* arr, int n);
```

22. Write a C++ function to copy one string to another string using pointer.

```
void StringCopy(char* str, char* &des_str);
```

23. Write a C++ function to concatenate two strings using pointer.

```
void StringConcate(char* s1, char* s2, char*& des_str);
```

24. Write a C++ function to print the elements of the array in reverse order using a pointer.

```
void PrintReverse(int* arr, int n);
```

25. A polynomial $f(x) = a_0x^0 + a_1x^1 + \cdots + a_nx^n$, where a_0 is a coefficient, is represented by the following structure:

```
struct Poly {
    int n;  //May be changed during computation
    int* a; //Should be allocated dynamically
};
```

Implement function `AddPoly` to perform $h = f + g$

```
int AddPoly(Poly f, Poly g, Poly & h);
```

26. Write a C++ function to get all starting and ending index of all subarrays that has sum of elements equals to S.

```
int** SegmentList(int* arr, int n, int S, int& subarr_size);
```

The function returns an array of list of indices.

Example: Input array: {1, 2, 3, 2, 2, 2, 3, 1, 5}, S = 6

➔ Return list: { (0, 2), (3, 5), (5, 7), (7,8) }, subarr_size = 4