Char and String functions

To compute a substring of a given string str, the substr() function can be used. The first argument of the function is the starting index of the substring, and the second argument is the length of the substring.

Here are few examples (using str = "Hello World"):

- str.substr(1) returns a substring of str starting from index 1 till the end of the string. "ello World
- str.substr(1, 3) returns a substring of str starting from index 1 and of length 3. "ell
- substr(1, str.length()-2) returns a substring of str starting from index 1 and of length
- str.length()-2. "ello Worl

The following functions can be used to check if a char value is a digit, letter, etc. You need to include the header fille <cctype> to use these functions.

- isalnum(c) checks if c is alphanumeric (either a digit or a letter)
- isalpha(c) checks if c is a letter
- isdigit(c) checks if c is a digit
- islower(c) checks if c is a lowercase letter
- isupper(c) checks if c is an uppercase letter

C-Style String Functions:

The strcpy and strcat functions from <cstring> are used to manipulate C-style strings.

1. **strcpy(destination, source)**
   Copies the content of source into destination. Ensure destination has enough memory to store the copied string, including the null terminator (\0). For example,

```
#include <iostream>
#include <cstring>

int main() {
   char source[] = "Hello";
   char destination[20];

   strcpy(destination, source); // Copy "Hello" into destination
   std::cout << "Copied string: " << destination << std::endl; // Output: Hello

   return 0;
}
```

2. **strcat(destination, source)**
   Appends the content of source to the end of destination. Ensure destination has

enough memory to store the combined strings, including the null terminator. For example,

```
#include <iostream>
#include <cstring>

int main() {
   char str1[20] = "Hello";
   char str2[] = " World";

   strcat(str1, str2); // Append " World" to "Hello"
   std::cout << "Concatenated string: " << str1 << std::endl; // Output: Hello World

   return 0;
}
```

3. **strlen(str)**
   The strlen function from <cstring> calculates the length of a C-style string (character array) by counting the characters until it encounters the null terminator (\0). It's often used when working with raw character arrays (char[]) to determine their length.

Lab Questions:

1. Write a program that dynamically allocates memory for an array of `n` integers, where `n` is provided by the user. Populate the array with values entered by the user, calculate the sum of the array elements, and deallocate the memory.
2. Create a program that uses dynamic memory allocation to create a 2D array (matrix). Take the number of rows and columns as input from the user, populate the matrix with random numbers, print the matrix, and then deallocate the memory.
3. Write a program that dynamically allocates memory for two strings entered by the user. Concatenate the two strings into a dynamically allocated result string and display it. Free all allocated memory after use.
4. Dynamically allocate memory for a 2D array, take user input for the elements, and compute the transpose of the matrix. Print both the original and transposed matrices.
5. Write a program to dynamically allocate memory for an array of `n` integers. Sort the array using the bubble sort algorithm and display the sorted array. Deallocate memory after sorting.
6. Write a C++ program that dynamically allocates memory for a string entered by the user and checks whether the string is a palindrome. A palindrome is a string that reads the same backward as forward (e.g., "radar", "level").