🧠 **Task 3: Dynamic Array with Pointer and Manual Memory Management**

Task 3 covers a basic yet important concept of dynamic memory allocation in C++ using pointers.

🧑‍💻 What this program does:

- Allocates a dynamic array using `new`

- Checks if user wants more size and reallocates if needed

- Accepts input values from the user

- Displays array values through a separate function

- Deallocates memory at the end using `delete[]`

💡 Purpose:

This task helps reinforce:

- Understanding of dynamic memory

- Pointer handling in arrays

- Clean memory practices (avoiding memory leaks)

📸 Optional: You can upload an output screenshot named output_task3.png alongside this file.

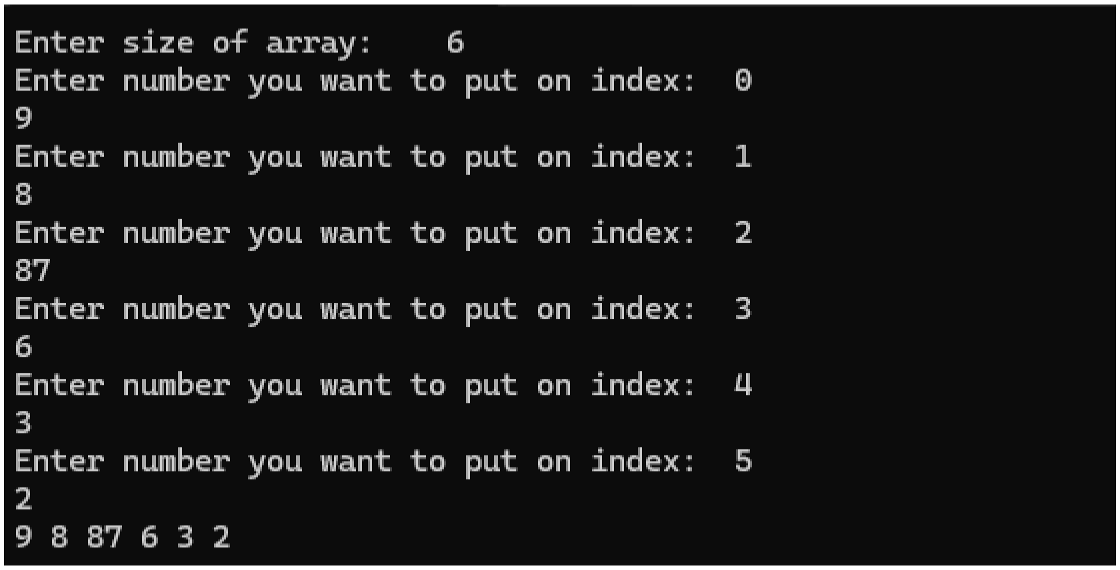
By practicing and uploading each of these tasks, I'm creating a clean archive of my learning journey — both for myself and anyone who might want to learn through code examples that are real and beginner-friendly.

Source code:

```
#include<iostream>

using namespace std;


// Function to print the array values

void print(int* ptr, int s){

	for(int i=0; i < s; i++){

		cout << ptr[i] << " ";

	}

}


int main(){

	int c = 5;


	// Dynamically allocating memory for 5 integers initially

	int* num = new int[c];


	int n;


	// Asking user for the actual size of the array they want

	cout << "Enter size of array:\t";

	cin >> n;


	// If the entered size is more than 5, we delete the previous memory and allocate new
one
```

```
	if(n > c){
		delete[] num; // freeing the previously allocated memory
		num = new int[n]; // allocating new memory with required size
	}

	// Taking user input and storing it in dynamically allocated array
	for(int i = 0; i < n; i++){
		cout << "Enter number you want to put on index:\t" << i << "\n";
		cin >> num[i];
	}

	// Calling function to print the array elements
	print(num, n);

	// After work is done, deallocating the memory to avoid memory leaks
	delete[] num;

	return 0;
}
```

Output screenshot:



```
Enter size of array:    6
Enter number you want to put on index:  0
9
Enter number you want to put on index:  1
8
Enter number you want to put on index:  2
87
Enter number you want to put on index:  3
6
Enter number you want to put on index:  4
3
Enter number you want to put on index:  5
2
9 8 87 6 3 2
```

Thanks & regards to MIKSI (github: miksi0078)