

Day-3

DOMAIN WINTER WINNING CAMP

Name : Yash Kumar

UID:22BCS15424

Section:620-B

1. Simple Calculator in C++ for Addition, Subtraction, Multiplication:

```
#include <iostream> using namespace std;
```

```
void calculator() {
```

```
    char operation;    float
```

```
    num1, num2;
```

```
    cout << "Enter operator (+, -, *): ";
```

```
    cin >> operation;
```

```
    cout << "Enter two operands: ";
```

```
    cin >> num1 >> num2;
```

```
    switch (operation) {
```

```
    case '+':
```

```
        cout << num1 << " + " << num2 << " = " << num1 + num2;
```

```
    break;    case '-':
```

```
        cout << num1 << " - " << num2 << " = " << num1 - num2;
```

```
    break;    case '*':
```

```
        cout << num1 << " * " << num2 << " = " << num1 * num2;
```

```
    break;    default:    cout << "Error! Operator is not
```

```
correct.";    break;
```

```

    }}

int main() {
    calculator();    return
0;
}

```

Output:

```

Simple Calculator
-----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 3
Enter two numbers for multiplication: 23
45
Result: 1035

```

2. C++ Program to Check Palindrome Using Function:

```
#include <iostream> using namespace std;
```

```

bool isPalindrome(int num) {    int original =
num, reversed = 0, remainder;    while (num
!= 0) {        remainder = num % 10;
reversed = reversed * 10 + remainder;
num /= 10;
    }

    return original == reversed;
}

```

```

int main() {    int number;    cout <<
"Enter a number: ";    cin >> number;    if
(isPalindrome(number))        cout <<
number << " is a palindrome.";    else

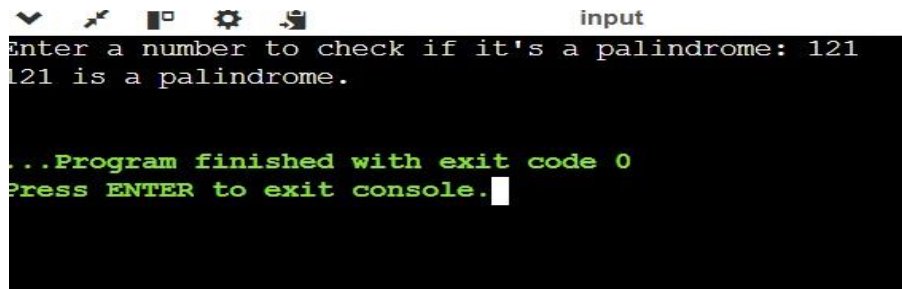
```

```

        cout << number << " is not a palindrome.";
return 0;
}

```

Output:



```

input
Enter a number to check if it's a palindrome: 121
121 is a palindrome.

...Program finished with exit code 0
Press ENTER to exit console.

```

3. Sum of Natural Numbers Using Recursion in C++:

```

#include <iostream> using namespace std;

int sumOfNaturalNumbers(int n) {
    if (n <= 1)        return n;

    return n + sumOfNaturalNumbers(n - 1);
}

int main() {
    int number;    cout << "Enter a
positive integer: ";    cin >> number;

    cout << "Sum of natural numbers up to " << number << " is " <<
sumOfNaturalNumbers(number);

    return 0;
}

```

Output :

```
Enter a positive integer: 23
The sum of the first 23 natural numbers is: 276

...Program finished with exit code 0
Press ENTER to exit console.█
```

4. Sum of Array Elements Using Recursion in C++:

```
#include <iostream> using namespace std;
```

```
int sumArray(int arr[], int size) {    if (size <=
0)        return 0;    return arr[size - 1] +
sumArray(arr, size - 1);
}

int main() {
int n;

    cout << "Enter the number of elements in array: ";

cin >> n;    int arr[n];

    cout << "Enter the elements of the array: ";
    for (int i = 0; i < n; i++)

cin >> arr[i];

    cout << "Sum of array elements: " << sumArray(arr, n);

return 0;

}
```

Output:

```
Enter the size of the array: 3
Enter the elements of the array: 2 3 9
The sum of the array elements is: 14

...Program finished with exit code 0
Press ENTER to exit console.█
```

5. Remove Linked List Element in C++:

```
#include <iostream> using
namespace std;

struct Node {
    int data;
    Node* next;
    Node(int x) : data(x), next(NULL) {}
};

Node* removeElement(Node* head, int val) {
    Node* dummy = new Node(0);    dummy-
>next = head;    Node* prev = dummy;
    while (head != NULL) {        if (head->data
== val) {            prev->next = head->next;
        } else {
            prev = head;
        }
        head = head->next;
    }
    return dummy->next;
}

void printList(Node* node) {
    while (node != NULL) {
        cout << node->data << " ";
        node = node->next;
    }
}
```

```

    }
}

int main() {
    Node* head = new Node(1);    head->
    next = new Node(2);    head->next->next
    = new Node(3);    head->next->next->next
    = new Node(4);

    cout << "Original list: ";
    printList(head);

    int valueToRemove = 3;    head =
    removeElement(head, valueToRemove);

    cout << "\nList after removing element " << valueToRemove << ": ";
    printList(head);

    return 0;
}

```

Output:

```

Original list: 1 2 3 4
List after removing element 3: 1 2 4

...Program finished with exit code 0
Press ENTER to exit console.

```

6. Find the Winner of Circular Game in C++:

```

#include <iostream>

#include <vector>

using namespace std;

int findWinner(int n, int k)
{
    vector<int> circle;
    for (int i = 1; i <= n; i++)
        circle.push_back(i);

    int index = 0;    while
(circle.size() > 1) {
        index = (index + k - 1) % circle.size();
        circle.erase(circle.begin() + index);
    }    return
circle[0];
}

int main() {    int
n = 5, k = 3;

    cout << "The winner of the circular game is: " << findWinner(n, k);

    return 0;
}

```

```

Enter the number of participants (n): 2
Enter the step size (k): 4
The winner is at position: 1

...Program finished with exit code 0
Press ENTER to exit console.

```

Ques 1. Sum of two numbers using function.

```
#include <iostream>

using namespace std; int
sumof(int x, int y)
{
return (x + y);
}

int main() { int a, b, sum = 0; cout <<
"Enter the first number: "; cin >> a;
cout << "Enter the second number: ";
cin >> b; sum = sumof(a, b);
cout << "Sum of " << a << " and " << b << " is: " << sum << endl; return
0;
}
```

Ques 2. array sum

```
#include <iostream>

#include <vector> using
namespace std;
vector<int>
sumArrays(const
vector<int>& a1, const
vector<int>& a2) {
vector<int> result;
```



```

int size = min(a1.size(), a2.size()); // Ensure the arrays align by size
for (int i = 0; i < size; i++) { result.push_back(a1[i] + a2[i]);
} return
result;
}

int main() { int n1, n2; cout << "Enter the number of elements
for the first array (a1): "; cin >> n1; vector<int> a1(n1);
cout << "Enter the elements of the first array (a1): ";
for (int i = 0; i < n1; i++) { cin >> a1[i];
}
cout << "Enter the number of elements for the second array (a2): ";
cin >> n2; vector<int> a2(n2);
cout << "Enter the elements of the second array (a2): ";
for (int i = 0; i < n2; i++) { cin >> a2[i];
}
vector<int> result = sumArrays(a1, a2);
cout << "Output array: ["; for (size_t i =
0; i < result.size(); i++) { cout <<
result[i]; if (i < result.size() - 1) { cout
<< ", ";
}
}
cout << "]" << endl; return
0;
}

```

Ques 3. Given the head of singly linked list reverse the list and return the reversed list

```
#include <iostream>

using namespace std;

struct Node { int
data;
Node* next;
};

Node* reverseList(Node* head) {
Node* prev = nullptr;
Node* current = head;
Node* next = nullptr;
while (current != nullptr) {
next = current->next;
current->next = prev; prev
= current; current = next;
} return
prev;
}

void printList(Node* head) {
while (head != nullptr) { cout
<< head->data << " "; head =
head->next;
}
cout << endl;
}
```

```

void appendNode(Node*& head, int data) {
Node* newNode = new Node(); newNode->
data = data; newNode->next = nullptr; if
(head == nullptr) {
head = newNode;
} else {
Node* temp = head; while
(temp->next != nullptr) { temp
= temp->next;
}
temp->next = newNode;
}
}

int main() { Node* head = nullptr; int n, value; cout
<< "Enter the number of elements in the list: "; cin
>> n; cout << "Enter the elements of the list: "; for
(int i = 0; i < n; i++) { cin >> value;
appendNode(head, value);
}

cout << "Original List: ";
printList(head); head =
reverseList(head); cout
<< "Reversed List: ";
printList(head); return 0;
}

```

----- write a function to
check if a number is prime or not

```

#include <iostream>

using namespace std;

bool isPrime(int n) {
    if (n <= 1) { return
false;
    } for (int i = 2; i * i <= n; i++)
    { if (n % i == 0) { return
false;
    }
    } return
true;
}

int main() { int
num;

cout << "Enter a number: ";
cin >> num; if
(isPrime(num)) {
cout << num << " is a prime number." << endl;
} else {
cout << num << " is not a prime number." << endl;
}

return 0;
}

```

Ques 4. write a function to reverse a string

```

#include <iostream> #include
<string> using namespace std;

```

```

string reverseString(string str) {
    int n = str.length(); for (int i =
    0; i < n / 2; i++) { swap(str[i],
    str[n - i - 1]);
    }
    return str;
}

int main() { string input;
    cout << "Enter a string: ";
    getline(cin, input);
    string reversed = reverseString(input); cout <<
    "Reversed string: " << reversed << endl; return
    0;
}

```

Ques 5. Implement a function that swap two variable using pass by reference

```

#include <iostream> using
namespace std; void
swapValues(int& a, int& b) { int
temp = a; a = b;
b = temp;
}
int main() { int
x, y;
    cout << "Enter two numbers: "; cin >> x >> y; cout <<
    "Before swapping: x = " << x << ", y = " << y << endl;
    swapValues(x, y);
}

```

```
cout << "After swapping: x = " << x << ", y = " << y << endl; return  
0;  
}
```

Ques 6. write a recursive function to compute the GCD of two numbers

```
#include <iostream>  
  
using namespace std;  
  
int gcd(int a, int b) {  
    if (b == 0) return a;  
    return gcd(b, a % b);  
}  
  
int main() {  
    int num1, num2;  
  
    cout << "Enter two numbers: ";  
  
    cin >> num1 >> num2; int  
    result = gcd(num1, num2);  
  
    cout << "GCD of " << num1 << " and " << num2 << " is: " << result << endl; return  
    0;  
}
```

Ques 7. Check weather number is perfect or not

```
#include <iostream> using  
namespace std; bool  
isPerfectNumber(int num) { if  
(num <= 1) return false; int sum
```

```
= 0; for (int i = 1; i <= num / 2;
++i) { if (num % i == 0) { sum
+= i;
}
}
return sum == num;
}

int main() { int num; cout << "Enter a number:
"; cin >> num; if (isPerfectNumber(num)) {
cout << num << " is a perfect number." << endl;
} else {
cout << num << " is not a perfect number." << endl;
}
return 0;
}
```