Name – Nandita Dutta

UID – 22BCS15313

Section – 620-A

DOMAIN WINTER WINNING CAMP

Ques 1 Factorial Of Number Using Recursion

```cpp
#include <iostream>

using namespace std;

int main() {
    const int size = 3;
    int a[size] = {2, 4, 3};
    int b[size] = {5, 4,
    5}; int result[size];

    // Adding corresponding elements of the two
    arrays for (int i = 0; i < size; ++i) {
        result[i] = a[i] + b[i];
```

```cpp
    }

    // Printing the result array with proper

    formatting for (int i = 0; i < size; ++i) {

        cout << result[i];

        if (i != size - 1) { // Add a comma and space except for the last

            element cout << ", ";

        }

    }


    return 0;

}
```

Ques 2

```cpp
#include

<iostream>  using

namespace    std;

int sum(int a , int

b){

    return a+b;

}

int main() {

    int ans= sum(2,3);

    cout<<"Sum is - "<<ans;


    return 0;

}
```

Ques 3

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

int main() {
    const int size = 3;
    int a[size] = {2, 4, 3};
    int b[size] = {5, 4, 5};

    int num1 = 0, num2 = 0;
    for (int i = 0; i < size;
        ++i) { num1 = num1
        * 10 + a[i]; num2 =
        num2 * 10 + b[i];
    }

    int sum = num1 +
    num2; cout << "Sum is:
    " << sum;
```

```cpp
    return 0;

}
```

Ques 4

```cpp
#include <iostream>

using namespace std;


struct Node
  { int data;

  Node*

  next;

};


Node* reverseLinkedList(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;

}
```

```cpp
void printList(Node*
head) { Node* temp =
head;
while (temp != nullptr) {
```

```cpp
        cout << temp->data << "

        "; temp = temp->next;

    }

    cout << endl;

}


void push(Node** head, int

    data) { Node* newNode =

    new Node(); newNode-

    >data = data; newNode-

    >next = *head;

    *head = newNode;

}


int main() {

    Node* head = nullptr;


    push(&head, 4);

    push(&head, 3);

    push(&head, 2);

    push(&head, 1);


    cout << "Original list: ";

    printList(head);


    head = reverseLinkedList(head);


    cout << "Reversed list:
```

"; printList(head);

```cpp
    return 0;
}
```

Ques 5

```cpp
#include<iostream>

using namespace std;


void prime(int

  n) { int count

  = 0;

  for(int i = 1; i <= n; i++)

    { if(n % i == 0) {

      count++;

    }

  }

  if(count == 2) {

    cout << "Prime number ";

  } else {

    cout << "Not a prime number ";

  }

}


int main() {

  int n;

  cout << "Enter a

  number: "; cin >> n;

  prime(n);

  return 0;

}
```

Ques 6

```cpp
#include<iostream>
#include<string>
#include<algorithm>
using namespace std;

int main() {
    string str;
    cout << "Enter a string:
    ";  cin >> str;
    reverse(str.begin(),
    str.end());

    cout << "Reversed string: " << str <<
    endl; return 0;
}
```

Ques 7

```cpp
#include<iostream>
using namespace std;
void gcd( int n1,int
n2){
    while(n1!=n2){
        if(n1>n2){
            n1=n1-
            n2;} else{
            n2=n2-
            n1;
```

```cpp
        }
    }
    cout<< "gcd - "<<n1;
}
int main(){
```

```cpp
    int n1,n2;

    cout<<"enter two numbers - ";

    cin>>n1>>n2;

    gcd(n1,n2); return
```

0; } **Ques 8**

```cpp
#include<iostream>

using namespace std;


void swap(int &a, int

    &b) { int c = a;

    a = b;

    b = c;

    cout << "Swapped numbers: " << a << " " << b << endl;

}


int main() {

    int a, b;

    cout << "Enter two

    numbers: "; cin >> a >> b;

    swap(a,

    b); return

    0;

}
```

Ques 9

```cpp
#include <iostream>

using namespace std;


bool isPerfectNumber(int
```

num) { if (num <= 1)

return false;

```cpp
    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;
}
```