Name – Abhay Bansal
UID- 22BCS15306
Date-23/12/2024
Section – 620 /A

Question 1

```
#include <iostream>
using namespace std;
void addArrays(int arr1[], int arr2[], int result[], int size) {
   for (int i = 0; i < size; i++) {        result[i] = arr1[i] +
arr2[i];
   }
}
int main() {     int size = 3;      int
arr1[size] = {1, 2, 3};     int arr2[size]
= {4, 5, 6};     int result[size];
   addArrays(arr1, arr2, result, size);
   cout << "Resultant array: ";     for (int i = 0; i
< size; i++) {
      cout << result[i] << " ";
   }
   return 0;
}
```

QUESTION 2

```
#include <iostream>
using namespace std
int addNumbers(int a, int b) {
   return a + b;
}

int main() {
   int num1, num2;
   cout << "Enter the first number: ";     cin >> num1;
   cout << "Enter the second number: ";     cin >> num2;
   int sum = addNumbers(num1, num2);
   cout << "The sum is: " << sum << endl;
   return 0;
}
```

QUESTION 3

```
#include <iostream>
using namespace std;
struct Node {     int
data;
   Node* next;
   Node(int val) : data(val), next(nullptr) {}
};

Node* reverseLinkedList(Node* head) {
   Node* prev = nullptr;     Node* curr =
head;     while (curr) {        Node* next =
curr->next;        curr->next = prev;
prev = curr;        curr = next;
   }     return prev;
}
void printList(Node* head) {     while (head)
{        cout << head->data << " ";
       head = head->next;
   }
   cout << endl;
}
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);
   head = reverseLinkedList(head);
   cout << "Reversed list: ";
   printList(head);
   return 0;
}
```

QUESTION 4

```
#include <iostream>
#include <string>
#include <algorithm>

void reverseString(std::string& str) {     std::reverse(str.begin(), str.end());
```

```
}

int main() {     std::string input;     std::cout
<< "Enter a string: ";
   std::getline(std::cin, input);

   reverseString(input);
   std::cout << "Reversed string: " << input << std::endl;

   return 0;
}
```

QUESTION 5

```
#include <iostream>

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

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

   if (isPrime(number)) {
      std::cout << number << " is a prime number." << std::endl;
   } else {
      std::cout << number << " is not a prime number." << std::endl;     }
```

```
   return 0;
}
```

QUESTION 6

```
#include <iostream>

int gcd(int a, int b) {     if (b ==
0) {
       return a; // Base case: GCD of (a, 0) is a
     }     return gcd(b, a % b); // Recursive case: gcd(a, b) = gcd(b, a % b) }

int main() {     int num1, num2;     std::cout <<
"Enter two numbers: ";
   std::cin >> num1 >> num2;

   std::cout << "GCD of " << num1 << " and " << num2 << " is
" << gcd(num1, num2) << std::endl;

   return 0;
}
```

QUESTION 7

```
#include <iostream>

void swap(int& a, int& b) {
   int temp = a;     a =
b;     b = temp;
}
```

```
int main() {
     int x, y;
   std::cout << "Enter two numbers: ";
   std::cin >> x >> y;

   std::cout << "Before swapping: x = " << x << ", y = " << y << std::endl;

   swap(x, y);

   std::cout << "After swapping: x = " << x << ", y = " << y << std::endl;

   return 0;
}
```

QUESTION 8

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

bool isPerfect(int num) {     if
(num <= 1) {        return false;
   }

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

   return sum == num;
}

int main() {     int
number;
```

```
    cout << "Enter a number: ";     cin >>
number;

    if (isPerfect(number)) {
        cout << number << " is a perfect number." << endl;
    } else {
        cout << number << " is not a perfect number." << endl;
    }
    return 0;
}
```

QUESTION 9

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

int fibonacci(int n) {

    if (n <= 1) {        return n;
    }

    return fibonacci(n - 1) + fibonacci(n - 2);
}

int main() {
    int n;

    cout << "Enter the number of terms: ";
    cin >> n;

    cout << "Fibonacci Series up to " << n << " terms:" << endl;

    for (int i = 0; i < n; i++) {
        cout << fibonacci(i) << " ";
    }
    cout << endl;

    return 0;
}
```