

1)Reverse the number using do..while loop:

```
#include <iostream>

#include <vector>

class Solution {
public:

    static std::vector<int> reverseNumber(int n) {

        std::vector<int> result;

        int reversed = 0;

        do

        {

            reversed=reversed*10+(n% 10);

            n/=10;

        }

        result.push_back(reversed);

        return result;

    }while(n>0);

};


int main() {

    int n = 1234;

    std::vector<int> result = Solution::reverseNumber(n);
```

```
std::cout << "Reversed number: " << result[0] << std::endl;

return 0;

}
```

2) Multiplication using for loop:

```
#include <iostream>

#include <vector>

class Solution {
public:

    static std::vector<int> printMultiplicationTable(int n) {

        std::vector<int> result;

        for(int i=1;i<=10;i++)

        {

            result.push_back(n*i);

        }

        return result;

    }

};

int main() {

    int n = 5;

    std::vector<int> result = Solution::printMultiplicationTable(n);
```

```

std::cout << "Multiplication table of " << n << ":" << std::endl;
for (int i = 0; i < result.size(); ++i) {

    std::cout << n << " x " << i + 1 << " = " << result[i] <<
        std::endl;

}

return 0;

}

```

3)Unique Elements in the given arrays using do..while loop.

```

#include <iostream>

#include <vector>

class Solution {
public:

    static std::vector<int> printUniqueElements(int arr[], int n) {

        std::vector<int> result;

        int i=0,j,count;

        do

        {

            j=0;

            count=0;

            do

            {

```

```

        if(arr[i]==arr[j]) count++;

        j++;
    }while(j<n);
    if(count==n)
    {
        result.push_back(-1);
        return result;
    }
    if(count==1) result.push_back(arr[i]);
    i++;
}while(i<n);
return result;
}

};

int main() {
    int arr[] = {4, 5, 6, 7, 4, 5, 9, 10, 6};
    int n = sizeof(arr) / sizeof(arr[0]);
    std::vector<int> result = Solution::printUniqueElements(arr, n);
    std::cout << "Unique elements: ";
    for (int elem : result) {
        std::cout << elem << " ";
    }
}

```

```

    }

    std::cout << std::endl;

    return 0;
}

```

4)Find the given number is prime if prime means add the digit if not return -1 using do..while loop.

```

#include <iostream>

#include <vector>

class Solution {
public:

    static std::vector<int> sumIfPrime(int n) {

        std::vector<int> result;

        int sum = 0,i=2,count=0;

        do

        {

            if(n%i==0) count++;

            i++;

        }while(i<n/2);

        if(count==0)

        {

            do

```

```

        {
            sum+=n%10;

            n/=10;

        }while(n>0);

    }

    else sum=-1;

    result.push_back(sum);

    return result;

}

};

int main() {

    int n = 1231;

    std::vector<int> result = Solution::sumIfPrime(n);

    std::cout << "Result: " << result[0] << std::endl;

    return 0;

}

```