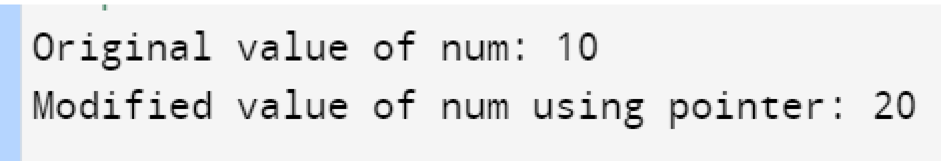
Stanzin Druksel 23103141

Assignment – 3 Pointers

Question 1)

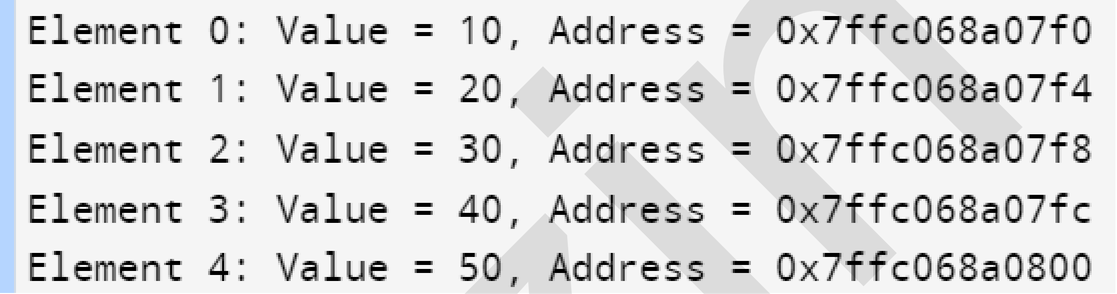
```
#include <iostream> using
namespace std; int main() {

   int num = 10;
   int* ptr = &num;
   cout << "Original value of num: " << num << endl;
   *ptr = 20;
   cout << "Modified value of num using pointer: " << num << endl;
return 0;}
```



```
Original value of num: 10
Modified value of num using pointer: 20
```

Question 2)

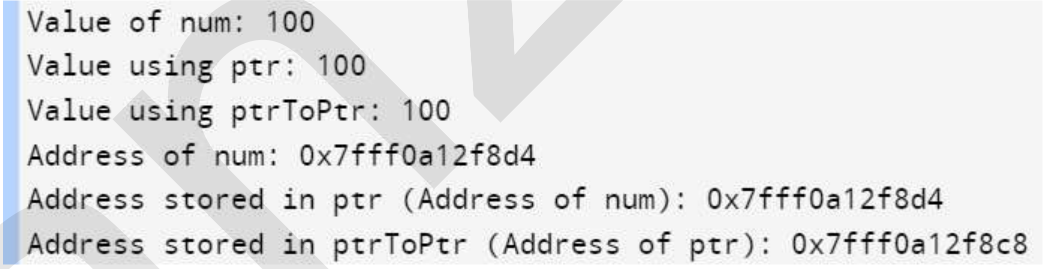
```
#include <iostream>
using namespace std;
int main() {
   int arr[5] = {10, 20, 30, 40, 50};
   int *ptr = arr;
   for(int i = 0; i < 5; i++) {
      cout << "Element " << i << ":
Value = " << *(ptr + i) << ", Address = " << (ptr + i) << endl;}
   return 0;}
```



```
Element 0: Value = 10, Address = 0x7ffc068a07f0
Element 1: Value = 20, Address = 0x7ffc068a07f4
Element 2: Value = 30, Address = 0x7ffc068a07f8
Element 3: Value = 40, Address = 0x7ffc068a07fc
Element 4: Value = 50, Address = 0x7ffc068a0800
```

Question 3)

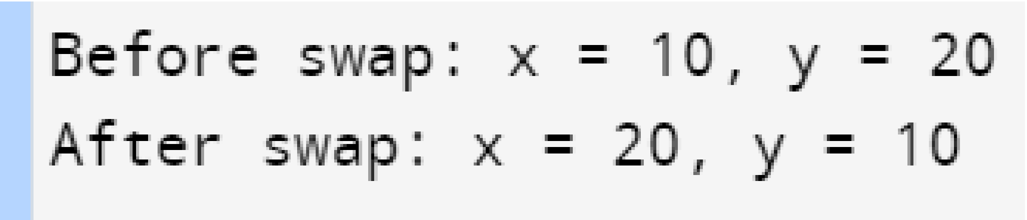
```
#include <iostream>
using namespace std;
int main() {
   int num = 100;
   int* ptr = &num;
   int** ptrToPtr = &ptr;
   cout << "Value of num: " << num << endl;
   cout << "Value using ptr: " << *ptr << endl;
   cout << "Value using ptrToPtr: " << **ptrToPtr << endl;
   cout << "Address of num: " << &num << endl;
   cout << "Address stored in ptr (Address of num): " << ptr << endl;
   cout << "Address stored in ptrToPtr (Address of ptr): " << ptrToPtr << endl;
   return 0;}
```



```
Value of num: 100
Value using ptr: 100
Value using ptrToPtr: 100
Address of num: 0x7fff0a12f8d4
Address stored in ptr (Address of num): 0x7fff0a12f8d4
Address stored in ptrToPtr (Address of ptr): 0x7fff0a12f8c8
```

Question 4)

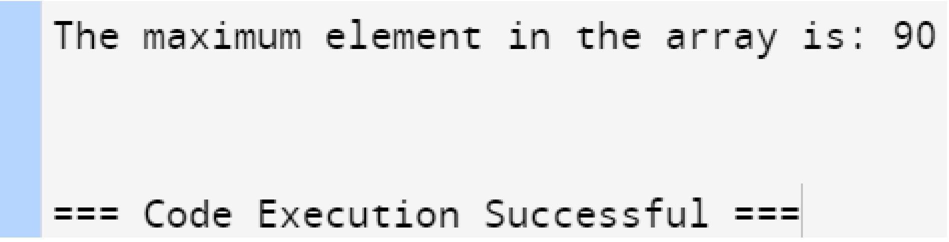
```
#include <iostream>
using namespace std; void swap(int* a, int* b) {
   int temp = *a;
   *a = *b;
   *b = temp;}
int main() {
   int x = 10;
   int y = 20;
   cout << "Before swap: x = " << x << ", y = " << y << endl;
   swap(&x, &y);
   cout << "After swap: x = " << x << ", y = " << y << endl;
   return 0;}
```



```
Before swap: x = 10, y = 20
After swap: x = 20, y = 10
```

Question 5)

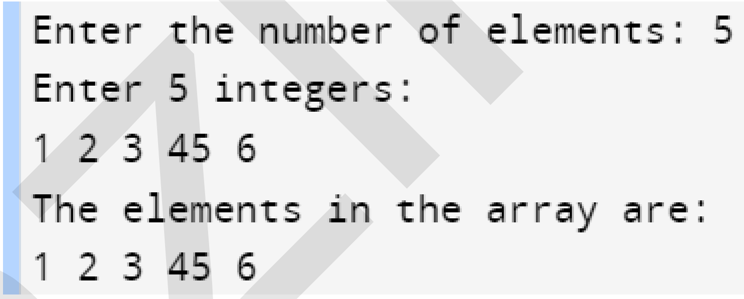
```
#include <iostream>
using namespace std;
int main() {
  int arr[] = {10, 50, 30, 90, 70};
  int n = sizeof(arr) / sizeof(arr[0]);
  int* ptr = arr;
  int max = *ptr;    for(int i = 1; i < n; i++) {
    if(*(ptr + i) > max) {
      max = *(ptr + i);}}
  cout << "The maximum element in the array is: " << max << endl;
  return 0;}
```



```
The maximum element in the array is: 90


=== Code Execution Successful ===
```

Question 6)

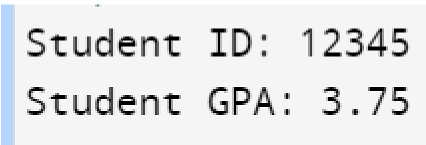
```
#include <iostream>
using namespace std;
int main() {
  int n;
  cout << "Enter the number of elements: ";
  cin >> n;
  int* arr = new int[n];
  if (arr == nullptr) {
    cout << "Memory allocation failed!" << endl;
    return 1;}
  cout << "Enter " << n << " integers:" << endl;     for (int i = 0; i < n;
i++) {
    cin >> arr[i];}
  cout << "The elements in the array are:" << endl;
  for (int i = 0; i < n; i++) {
    cout << arr[i] << " ";}
  cout << endl;
  delete[] arr;
  return 0;}
```



```
Enter the number of elements: 5
Enter 5 integers:
1 2 3 45 6
The elements in the array are:
1 2 3 45 6
```

Question 7)

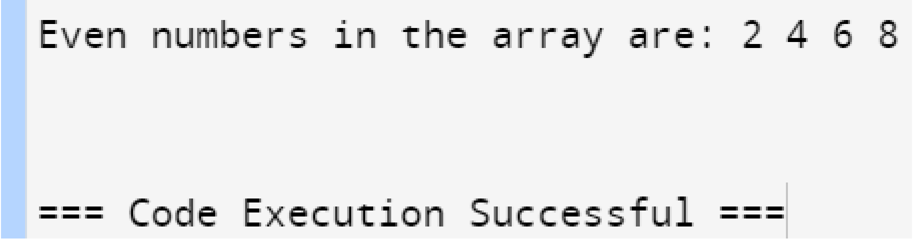
```
#include <iostream>
using namespace std;
struct Student {
  int id;
  float gpa;};
void printStudent(const Student* student) {
  cout << "Student ID: " << student->id << endl;
  cout << "Student GPA: " << student->gpa << endl;}
int main() {
  Student s1;
  s1.id = 12345;
  s1.gpa = 3.75;
  printStudent(&s1);
  return 0;}
```



```
Student ID: 12345
Student GPA: 3.75
```

Question 8)

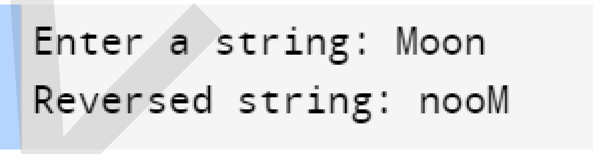
```
#include <iostream>
using namespace std;
int* extractEvenNumbers(int* arr, int size, int& evenCount) {
  evenCount = 0;
  for (int i = 0; i < size; i++) {
    if (arr[i] % 2 == 0) {
      evenCount++;}}
  int* evenArr = new int[evenCount];
  int j = 0;
  for (int i = 0; i < size; i++) {
    if (arr[i] % 2 == 0) {
      evenArr[j] = arr[i];
      j++;}}
  return evenArr;}
int main() {    int arr[] = {1, 2, 3, 4, 5, 6, 7, 8};
  int size = sizeof(arr) / sizeof(arr[0]);
  int evenCount = 0;
  int* evenArr = extractEvenNumbers(arr, size, evenCount);
  cout << "Even numbers in the array are: ";
  for (int i = 0; i < evenCount; i++) {
    cout << evenArr[i] << " ";}
  cout << endl;
  delete[] evenArr;
  return 0;}
```



```
Even numbers in the array are: 2 4 6 8



=== Code Execution Successful ===
```

Question 9)

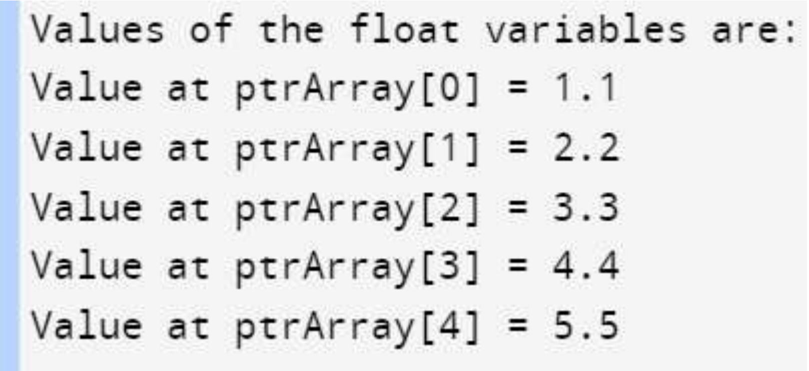
```
#include <iostream>
using namespace std;
void reverseString(char* str) {
    char* start = str;
    char* end = str;
    while (*end) {
        ++end;}
    --end;
    while (start < end) {
        char temp = *start;
        *start = *end;
        *end = temp;        ++start;
        --end;}}
int main() {
    char str[100];
    cout << "Enter a string: ";
    cin.getline(str, sizeof(str));
    reverseString(str);
    cout << "Reversed string: " << str << endl;
    return 0;}
```



```
Enter a string: Moon
Reversed string: nooM
```

Question 10)

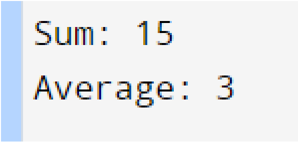
```
#include <iostream>
using namespace std;
int main() {
    float a = 1.1f, b = 2.2f, c = 3.3f, d = 4.4f, e = 5.5f;
    float* ptrArray[5] = {&a, &b, &c, &d, &e};
    cout << "Values of the float variables are:" << endl;
    for (int i = 0; i < 5; i++) {
        cout << "Value at ptrArray[" << i << "] = " << *ptrArray[i]
<< endl;}
    return 0;}
```



```
Values of the float variables are:
Value at ptrArray[0] = 1.1
Value at ptrArray[1] = 2.2
Value at ptrArray[2] = 3.3
Value at ptrArray[3] = 4.4
Value at ptrArray[4] = 5.5
```

Question 11)

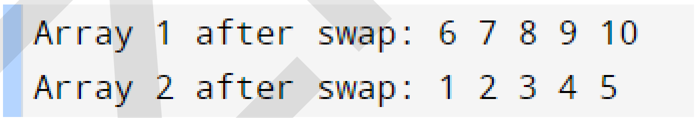
```
#include <iostream>
using namespace std;
void calculateSumAndAverage(int arr[], int size, int &sum, float &average) {
    sum = 0;
    for (int i = 0; i < size; i++) {
        sum += arr[i];}
    average = static_cast<float>(sum) / size;}
int main() {
    int arr[5] = {1, 2, 3, 4, 5};
    int sum;
    float average;
    calculateSumAndAverage(arr, 5, sum, average);
    cout << "Sum: " << sum << endl;
    cout << "Average: " << average << endl;
    return 0;}
```



```
Sum: 15
Average: 3
```

Question 12)

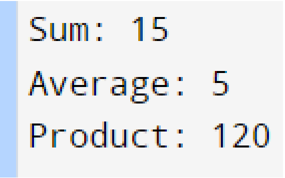
```
#include <iostream>
using namespace std;
void swapArrays(int* arr1, int* arr2, int size) {
    for (int i = 0; i < size; i++) {
        int temp = *(arr1 + i);
        *(arr1 + i) = *(arr2 + i);
        *(arr2 + i) = temp;}}
int main() {
    int arr1[] = {1, 2, 3, 4, 5};
    int arr2[] = {6, 7, 8, 9, 10};    int size = sizeof(arr1) / sizeof(arr1[0]);
    swapArrays(arr1, arr2, size);
    cout << "Array 1 after swap: ";
    for (int i = 0; i < size; i++) {
        cout << arr1[i] << " ";}
    cout << endl;
    cout << "Array 2 after swap: ";
    for (int i = 0; i < size; i++) {
        cout << arr2[i] << " ";}
    cout << endl;
    return 0;}
```



```
Array 1 after swap: 6 7 8 9 10
Array 2 after swap: 1 2 3 4 5
```

Question 13)

```
#include <iostream>
using namespace std;
void calculate(int a, int b, int c, int* sum, float* average, int* product) {
    *sum = a + b + c;
    *average = static_cast<float>(*sum) / 3;
    *product = a * b * c;}
int main() {
    int a = 4, b = 5, c = 6;
    int sum, product;
    float average;
    calculate(a, b, c, &sum, &average, &product);
    cout << "Sum: " << sum << endl;
    cout << "Average: " << average << endl;
    cout << "Product: " << product << endl;
    return 0;}
```

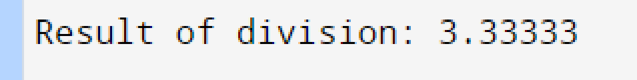


```
Sum: 15
Average: 5
Product: 120
```

Question 14)

```
#include <iostream>
using namespace std;
float divide(int a, int b) {
    if (b == 0) {
        cout << "Division by zero is undefined." << endl;
        return 0.0;}
```

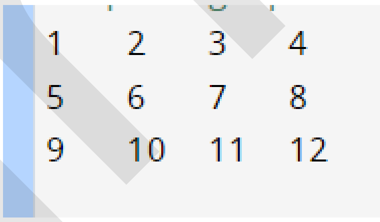
```
  return static_cast<float>(a) / b;}
int main() {
  float (*funcPtr)(int, int) = divide;
  int x = 10, y = 3;
  float result = funcPtr(x, y);
  cout << "Result of division: " << result << endl;
  return 0;}
```



```
Result of division: 3.33333
```

Question 15)

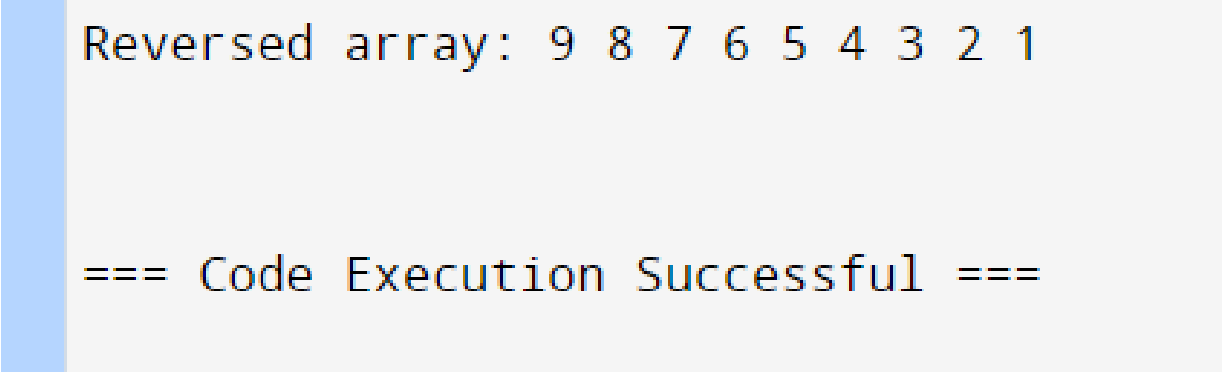
```
#include <iostream>
using namespace std;
int main() {
  int rows = 3, cols = 4;
  int** arr = new int*[rows];
  for (int i = 0; i < rows; i++) {
    arr[i] = new int[cols];}
  int value = 1;
  for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
      arr[i][j] = value++;}}
  for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
      cout << arr[i][j] << "\t";}
    cout << endl;}
  for (int i = 0; i < rows; i++) {
    delete[] arr[i];}
  delete[] arr;
  return 0;}
```



```
1    2    3    4
5    6    7    8
9    10   11   12
```

Question 16)

```
#include <stdio.h>
void reverseArray(int* arr, int size) {
  int *start = arr;
  int *end = arr + size - 1;
  while (start < end) {
    int temp = *start;
    *start = *end;
    *end = temp;
    start++;
    end--;}}
int main() {
  int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
  int size = sizeof(arr) / sizeof(arr[0]);
  reverseArray(arr, size);
  printf("Reversed array: ");
  for (int i = 0; i < size; i++) {
    printf("%d ", arr[i]);}
  printf("\n");
  return 0;}
```



```
Reversed array: 9 8 7 6 5 4 3 2 1


=== Code Execution Successful ===
```