Question 1: --------------------------------------[Source Code]-------------------------------------

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

*//proto to output the switch case options to choose from*void intro() {  
 *string* intro = "-----Choose Return Method:-----\n[1] value\n[2] reference\n[3] prototype";  
 cout **<<** intro **<<** endl;  
}  
*//return by prototype*void sqr\_proto(double *x*) {  
 double sqrX = (*x* \* *x*);  
 cout **<<** sqrX **<<** endl;  
}  
*//return by reference*int sqr\_ref(double &*x*) {  
 return *x* \* *x*;  
}  
int main() {  
 int method;  
 double x{0};  
 bool exit = false;  
 *string* user\_exit;  
 do {  
 intro();*//calls the prototype 'intro' to list return-method options* cin **>>** method;  
 cout **<<** "Enter a number: " **<<** endl;  
 cin **>>** x;  
 *//cleaner way of combining the three ways of squaring a number into one working program* switch (method) {  
 case 1: *//return by value* cout **<<** (x \* x) **<<** endl;  
 break;  
 case 2: *//return by reference* cout **<<** sqr\_ref(x) **<<** endl;  
 break;  
 case 3: *//for return by prototype* sqr\_proto(x);  
 break;  
 }  
 cout **<<** "Continue?(y/n): ";  
 cin **>>** user\_exit;  
 if (user\_exit **!=** "y") { exit = true; }  
 } while (exit != true);  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

-----Choose Return Method: -----

[1] value

[2] reference

[3] prototype

1

Enter a number:

9

81

Continue?(y/n):y

-----Choose Return Method:-----

[1] value

[2] reference

[3] prototype

2

Enter a number:

47

2209

Continue?(y/n):y

-----Choose Return Method:-----

[1] value

[2] reference

[3] prototype

3

Enter a number:

4.934

24.3444

Continue?(y/n):n

Process finished with exit code 0

#####################################################################################

#####################################################################################

Question 2: --------------------------------------[Source Code]-------------------------------------

-------------------------------------[OUTPUT]-------------------------------------

#####################################################################################

#####################################################################################

Question 3: --------------------------------------[Source Code]-------------------------------------

#include <iostream>  
#include <climits>  
using namespace std;  
int main() {  
 //[SOURCE]--> https://www.geeksforgeeks.org/int\_max-int\_min-cc-applications/  
 //using INT\_MAX & INT\_MIN from <climits>  
 //These two will block/only allow for the greatest(for INT\_MAX) and least(for INT\_MIN) to be stored in memory  
 float num\_min = INT\_MAX, num\_max = INT\_MIN;  
 float arr1[] = { -333.12, 324523, 4233.3376, 9, 2.22556, -356.843 };  
 //using 'auto' will accept any types of numbers from 'arr1' for 'i' that are previously declared  
 // By using a loop, we are comparing each value stored in 'i' from 'arr1'...  
 //...to what is currently held in 'INT\_MIN' until the end of the array has been reached.  
 for (auto i: arr1) {  
 if (i < num\_min) {num\_min = i;}  
 }  
 for (auto i : arr1){  
 if (i > num\_max) {num\_max = i;}  
 }  
 cout << "Minimum: " << num\_min << endl;  
 cout << "Maximum: " << num\_max << endl;  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

Minimum: -356.843

Maximum: 324523

Process finished with exit code 0

#####################################################################################

#####################################################################################

Question 4: --------------------------------------[Source Code]-------------------------------------

#include <iostream>  
#include <algorithm>  
using namespace std;  
//[SOURCE] --> https://www.tutorialspoint.com/cplusplus-program-to-implement-sorted-array  
int main()  
{  
 float arr[] = { 1.76, 7.33, 6.11, 9.76, 5.99, 12.22};  
 float arr\_median;  
 int numofElem = sizeof(arr) / sizeof(arr[0]);  
 sort(arr, arr + numofElem);  
  
 cout << "Array: {";  
 //for sorting 'arr' using recursion  
 for (int i = 0; i < numofElem; ++i){  
 cout << arr[i] << ", ";  
 }  
  
 //for median  
 if(numofElem % 2 != 0){  
 //'\b\b' remove the last comma from the ouput of the sorted 'arr'  
 cout << "\b\b}\nArray is odd"<< endl;  
 //outputs the middle number of 'arr'  
 arr\_median = arr[numofElem/2];  
 cout << "Median: [" << arr\_median << "]" << endl;  
 }  
 else{  
 cout << "\b\b}\nArray is even"<< endl;  
 //outputs the product of the two middle numbers of 'arr' divided by 2(avg or mean)  
 arr\_median = (arr[numofElem/2] + arr[(numofElem/2) - 1])/(2);  
 cout << "Median: [" << arr\_median << "]" << endl;  
 }  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

---For Even array----

Array: {1.76, 5.99, 6.11, 7.33, 9.76, 12.22}

Array is even

Median: [6.72]

Process finished with exit code 0

---For Odd array----

Array: {1.76, 5.99, 6.11, 7.33, 9.76}

Array is odd

Median: [6.11]

Process finished with exit code 0

#####################################################################################

#####################################################################################

#####################################################################################

#####################################################################################

Question 6: --------------------------------------[Source Code]-------------------------------------

[Sources] 🡪 [[1]](https://www.programiz.com/dsa/bubble-sort) [[2]](https://www.softwaretestinghelp.com/bubble-sort/)

#include <iostream>  
using namespace std;  
//[Source](for 'swap' info and implementation) --> https://www.programiz.com/dsa/bubble-sort  
void bubbleSort(float arr[], int num\_of\_elem);  
void bubbleSort(float \*arr, int num\_of\_elem) {  
 int i, j;  
 //using the built-in function 'swap' to compare two values, if condition is met, the values are then swapped positions.  
 //this will repeat sequentially(starting from comparing first and second value onwards) until the end of 'arr' has been reached.  
 //once the end of 'arr' has been reached, the loop will repeat and continue this until there are no more valid swap-arguments.  
 for (i = 0; i < num\_of\_elem; i++) {  
 for (j = 0; j < num\_of\_elem - i - 1; j++) {  
 if (arr[j] < arr[j + 1]) {  
 swap(arr[j], arr[j + 1]);  
 }  
 }  
 }  
}  
//mostly reused code  
int main() {  
 float arr[] = {1.76, 7.33, 6.11, 9.76, 5.99, 34.5, 6, 9, 0};  
 int num\_of\_elem = sizeof(arr) / sizeof(int);  
  
 bubbleSort(arr, num\_of\_elem);  
  
 cout << "Descending Array {";  
 // Print the Sorted Array  
 for (int i = 0; i < num\_of\_elem; i++) {  
 cout << arr[i] << ", ";  
 }  
 //for better output format  
 cout << "\b\b}" << endl;  
 return (0);  
}

-------------------------------------[OUTPUT]-------------------------------------

Descending Array {34.5, 9.76, 9, 7.33, 6.11, 6, 5.99, 1.76, 0}

Process finished with exit code 0

#####################################################################################

#####################################################################################

Question 7: --------------------------------------[Source Code]-------------------------------------\

[Sources] 🡪 [[1]](https://www.programiz.com/cpp-programming/pointers-arrays) [[2]](https://www.softwaretestinghelp.com/new-delete-operators-in-cpp/) [[3]](https://stackoverflow.com/questions/35532427/how-to-dynamically-allocate-arrays-in-c)

#include <iostream>  
#include <algorithm>  
#include <new>  
using namespace std;  
  
int main() {  
 //creating array & pointers  
 int num\_of\_elem;  
 int \*ptr;  
 ptr = new int();  
  
 cout << "Enter number of array elements[1,100]: ";  
 cin >> num\_of\_elem;  
 \*ptr = num\_of\_elem;  
 if (num\_of\_elem <= 100 && num\_of\_elem >= 1) {  
 float \*arr = nullptr;//clears any value/data from 'arr' by pointing to a 'null'/empty value. Is not used here specifically, though, would be useful depending on the implementation  
 arr = new float[num\_of\_elem]; //creates a placeholder for array 'arr' with size = 'num\_of\_elem'  
 for (int input\_elem\_cnt = 1, n = 0; n < num\_of\_elem; n++, input\_elem\_cnt++ ) {  
 cout << "Element[" << input\_elem\_cnt << "] = ";//to keep a live count for the user to know how many elements have been inputted  
 cin >> \*(arr + n); //stores number from input to memory  
 }  
 sort(arr, arr + num\_of\_elem);  
 cout << "Your Array: {";  
 for (int i = 0; i < num\_of\_elem; i++) {  
 cout << \*(arr + i) << ", "; //outputs the numbers which are pointed to by their values  
 }  
 cout << "\b\b}" << endl;  
 delete ptr, delete[] arr;  
 }  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

Enter number of array elements[1,100]: 10

Element[1] = 1

Element[2] = 78

Element[3] = 6

Element[4] = 4

Element[5] = 2

Element[6] = 67

Element[7] = -999

Element[8] = 222

Element[9] = 46.77770

Element[10] = 0

Your Array: {-999, 0, 1, 2, 4, 6, 46.7777, 67, 78, 222}

#####################################################################################

#####################################################################################

Question 8: --------------------------------------[Source Code]-------------------------------------

#include <iostream>  
#include <cmath>  
using namespace std;  
  
float sqr(float x){  
 float c = (x \* x);  
 return c;  
}  
float std\_dev(float x1,float x2, float x3, float x4){  
 float product;  
 float avg = ( x1 + x2 + x3 + x4 ) / 4;  
 //calculations  
 float tmp1 = x1 - avg;  
 tmp1 = sqr(tmp1);  
 float tmp2 = x2 - avg;  
 tmp2 = sqr(tmp2);  
 float tmp3 = x3 - avg;  
 tmp3 = sqr(tmp3);  
 float tmp4 = x4 - avg;  
 tmp4 = sqr(tmp4);  
 float tmp = tmp1 + tmp2 + tmp3 + tmp4;  
 product = tmp / 4;  
 product = sqrt(product);  
 cout << "Mean: " << avg << "\nStandard Deviation: " << product << endl;  
}  
  
int main() {  
 float x1, x2, x3, x4;  
 cout << "Enter 4 values: " << endl;  
 //input  
 cin >> x1 >> x2 >> x3 >> x4;  
 //mean  
 std\_dev(x1, x2, x3, x4);  
 return 0;  
  
  
  
}

-------------------------------------[OUTPUT]-------------------------------------

Enter 4 values:

98

43

687.6

4

Mean: 208.15

Standard Deviation: 278.818

#####################################################################################

#####################################################################################

Question 9: --------------------------------------[Source Code]-------------------------------------

#include <iostream>  
using namespace std;  
  
inline float sum\_of\_nums(float x1,float x2,float x3,float x4,float x5){  
 float sum = x1 + x2 + x3 + x4 + x5;  
 return sum;  
};  
inline float prod\_of\_nums(float x1,float x2,float x3,float x4,float x5){  
 float prod = x1 \* x2 \* x3 \* x4 \* x5;  
 return prod;  
};  
int main() {  
 float x1, x2, x3, x4, x5, sum, prod;  
 cout << "Enter Values: " << endl;  
 cin >> x1 >> x2 >> x3 >> x4 >> x5;  
 sum = sum\_of\_nums(x1, x2, x3, x4, x5);  
 prod = prod\_of\_nums(x1, x2, x3, x4, x5);  
 cout << "Sum: " << sum << "\nPorduct: " << prod << endl;  
 return 0;  
}

-------------------------------------[OUTPUT]-------------------------------------

6

9.332

43.5

-43

3

Sum: 18.832

Porduct: -314199

Process finished with exit code 0

#####################################################################################

#####################################################################################

Question 10: --------------------------------------[Source Code]-------------------------------------

#include <iostream>

#include <string>

using namespace std;

string reverse(string usr\_input){

for (int i = usr\_input.length(); i > 0; i--) {

cout << usr\_input[i - 1];

}

return usr\_input;

}

int main() {

string \*usr\_input;

usr\_input = new string;

cout << "Enter string to be reversed: ";

cin >> \*usr\_input;

reverse(\*usr\_input);

delete usr\_input;

return 0;

}-------------------------------------[OUTPUT]-------------------------------------

Enter string to be reversed: testing\_assignment2\_Q10

01Q\_2tnemngissa\_gnitset

Process finished with exit code 0

#####################################################################################

#####################################################################################