

PROBLEM SOLVING

(Solving various problems using C Language)

Summer Internship Report Submitted in partial fulfillment

of the requirement for under graduate degree of

Bachelor of Technology

In

Computer Science Engineering

By

MUTNURI SUSHMA

221710305033

(<https://github.com/MutnuriSushma333/>)

Under the Guidance of



Department Of Computer Science Engineering

GITAM School of Technology

GITAM (Deemed to be University)

Hyderabad-502329

July 2020

DECLARATION

I submit this industrial training work entitled “**SOLVING VARIOUS PROBLEMS USING C LANGUAGE**” to GITAM (Deemed To Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer Science Engineering**”. I declare that it was carried out independently by me under the guidance of **Mr.**, Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD

MUTNURI SUSHMA

Date:

221710305033



GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India.

Dated:

CERTIFICATE

This is to certify that the Industrial Training Report entitled **“SOLVING VARIOUS PROBLEMS USING C LANGUAGE”** is being submitted by MUTNURI SUSHMA(221710305033) in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science Engineering** at GITAM (Deemed To Be University), Hyderabad during the academic year 2019-20

It is faithful record work carried out by her at the **Computer Science Department**, GITAM University Hyderabad Campus under my guidance and supervision.

Mr.
Assistant Professor
Department of CSE

Dr.Phani Kumar
Professor and HOD
Department of CSE

ACKNOWLEDGEMENT

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank respected **Dr. N. Siva Prasad**, Pro Vice Chancellor, GITAM Hyderabad and **Dr. CH. Sanjay**, Principal, GITAM Hyderabad

I would like to thank respected **Dr. Phani Kumar**, Head of the Department of Computer Science Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present a internship report. It helped me a lot to realize of what we study for.

I would like to thank the respected faculties **Mr.** who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

MUTNURI SUSHMA
221710305033

TABLE OF CONTENTS

1 Introduction to the project	1
2 Problem 1 (Reservoir Sampling)	2
2.1 Problem Statement	2
2.2 Coding	3
2.3 Output	4
3 Problem 2 (The Stock Span Problem)	5
3.1 Problem Statement	5
3.2 Coding	7
3.3 Output	8
4 Problem 3 (A Product Array Puzzle)	9
4.1 Problem Statement	9
4.2 Coding	11
4.3 Output	12
5 Problem 4 (A pancake Sorting Problem)	13
5.1 Problem Statement	13
5.2 Coding	15
5.3 Output	16
6 Problem 5 (Bill Payment Receipt)	17
6.1 Problem Statement	17
6.2 Coding	20
6.3 Output	21
7 Software Requirements	22
7.1 Hardware Requirements	
7.2 Software Requirements	

References

1.INTRODUCTION TO THE PROJECT

Problem Solving is the Process of Designing and carrying out certain steps to reach a Solution. Five problems which are listed below are of different complexity and require different approach and logics in order to achieve desired Output/ Solution.

1. **Reservoir Sampling** - In this problem we generate random samples from the list of many items.
2. **The Stock span problem** - In this problem we aims to calculate the span of stock's price for all n days.
3. **A product array puzzle** - In this problem we aims to calculate the product of the given array.
4. **A pancake Sorting Problem-** In this problem we Aims to Sort the elements in an array by Flipping the elements in an array up to a point and then Sorting the list of elements.
5. **Bill Payment Receipt** - In this problem we enables us to generate the Bill Receipts in the Super market.

I have executed projects in C language and Python. For C language, I have used DEV C++ to execute the codes.

2. PROBLEM1:RESERVOIR SAMPLING

This project aims to develop a project which is helpful for choosing k samples from n number of items in a list.

2.1 Problem Statement:

Reservoir sampling is a family of randomized algorithms for randomly choosing k samples from a list of n items, where n is either a very large or unknown number. Typically n is large enough that the list doesn't fit into main memory. For example, a list of search queries in Google and Facebook.

So we are given a big array (or stream) of numbers (to simplify), and we need to write an efficient function to randomly select k numbers where $1 \leq k \leq n$. Let the input array be `stream[]`.

Concepts Used:

1. Random Algorithm: Randomization is a fundamental technique in algorithm design, that allows programs to run quickly when the average-case behaviour of an algorithm is better than the worst-case behaviour. It is also heavily used in games, both in entertainment and gambling. The latter application gives the only known example of a programmer being murdered for writing bad code.

2. Header file <time.h>: In C programming language **time.h** (used as **ctime** in C++) is a header file defined in the C Standard Library that contains time and date function declarations to provide standardized access to time/date manipulation and formatting.

3. Srand function: The `srand()` function sets the starting point for producing a series of pseudo-random integers. If `srand()` is not called, the `rand()` seed is set as if `srand(1)` were called at program start. Any other value for seed sets the generator to a different starting point.

Syntax: `void srand(unsigned rand)`

4. For loop: A For loop is a control flow statement for specifying iteration, which allows code to be executed repeatedly.

Syntax : `for(initialization,condition,increment)`

```
{  
    Statements;  
}
```


Test Cases:

Expected Output:

Following are k randomly selected items

1 10 6 7 12

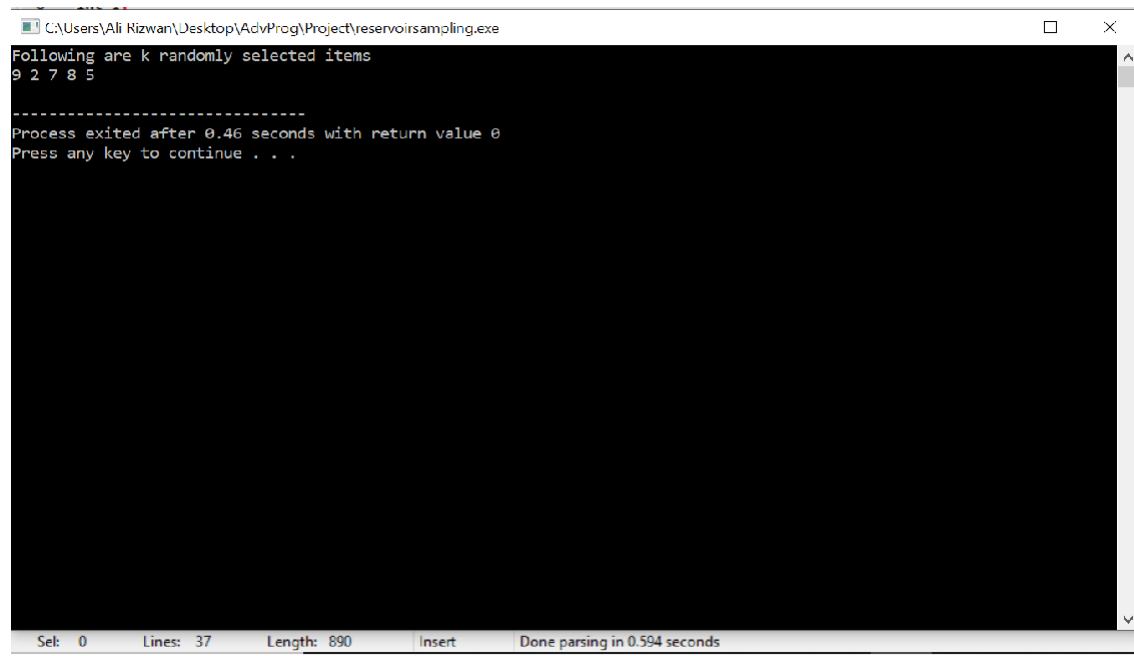
2.2 Coding:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int i;
5  void printarray(int stream[], int n)
6  {
7      for ( i = 0; i < n; i++)
8          printf("%d ", stream[i]);
9      printf("\n");
10 }
11 void selectKItems(int stream[], int n, int k)
12 {
13     int reservoir[k];
14     for (i = 0; i < k; i++)
15         reservoir[i] = stream[i];
16     srand(time(NULL));
17     for (; i < n; i++)
18     {
19         int j = rand() % (i+1);
20         if (j < k)
21             reservoir[j] = stream[i];
22     }
23     printf("Following are k randomly selected items \n");
24     printarray(reservoir, k);
25 }
26 int main()
27 {
28     int stream[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
29     int n = sizeof(stream)/sizeof(stream[0]);
30     int k = 5;
31     selectKItems(stream, n, k);
32     return 0;
33 }
```

Compile Log Debug Find Results

Fig 2.2.1

2.3 Output:



```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\reservoirsampling.exe
Following are k randomly selected items
9 2 7 8 5
-----
Process exited after 0.46 seconds with return value 0
Press any key to continue . . .

Sel: 0   Lines: 37   Length: 890   Insert   Done parsing in 0.594 seconds
```

Fig 2.3.1

3.PROBLEM 2: THE STOCK SPAN PROBLEM

This Project aims to calculate the span of stock's price for all n days.

3.1 Problem Statement:

The stock span Problem is a financial problem where we have a series of n daily price quotes for a stock and we need to calculate span of stock's price for all n days. The span S_i of the stock's price on a given day i is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than or equal to its price on the given day.

For example, if an array of 7 days prices is given as {100, 80, 60, 70, 60, 75, 85}, then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6}.

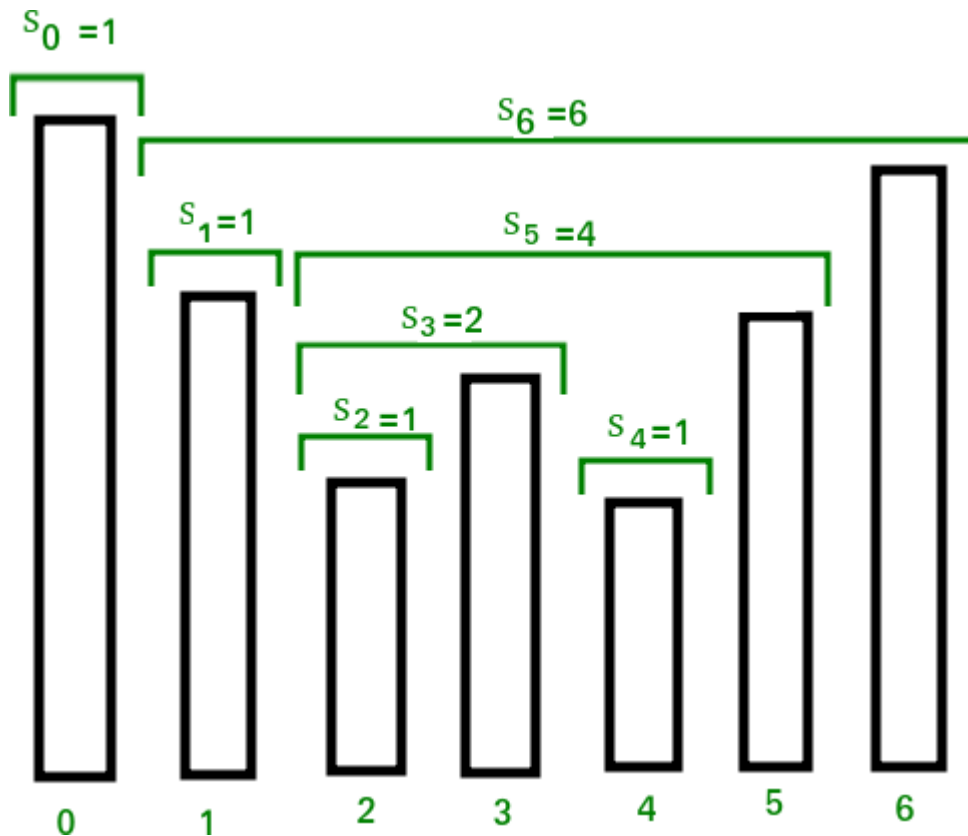


Fig 3.1.1

Concepts Used:

1. For loop: A For loop is a control flow statement for specifying iteration, which allows code to be executed repeatedly.

Syntax : for(initialization,condition,increment)

```
{  
    Statements;  
}
```

2. Logical Operator AND (&&): The logical AND operator (&&) returns the boolean value TRUE if both operands are TRUE and returns FALSE otherwise. ... The first operand is completely evaluated and all side effects are completed before continuing evaluation of the logical AND expression.

```
Syntax: if (a&&b)  {  
    Printf(“Logical And”);  
}
```

3. Arrays: An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

Syntax: int array[100];

Test Cases:

Expected Output:

1 1 2 4 5 1

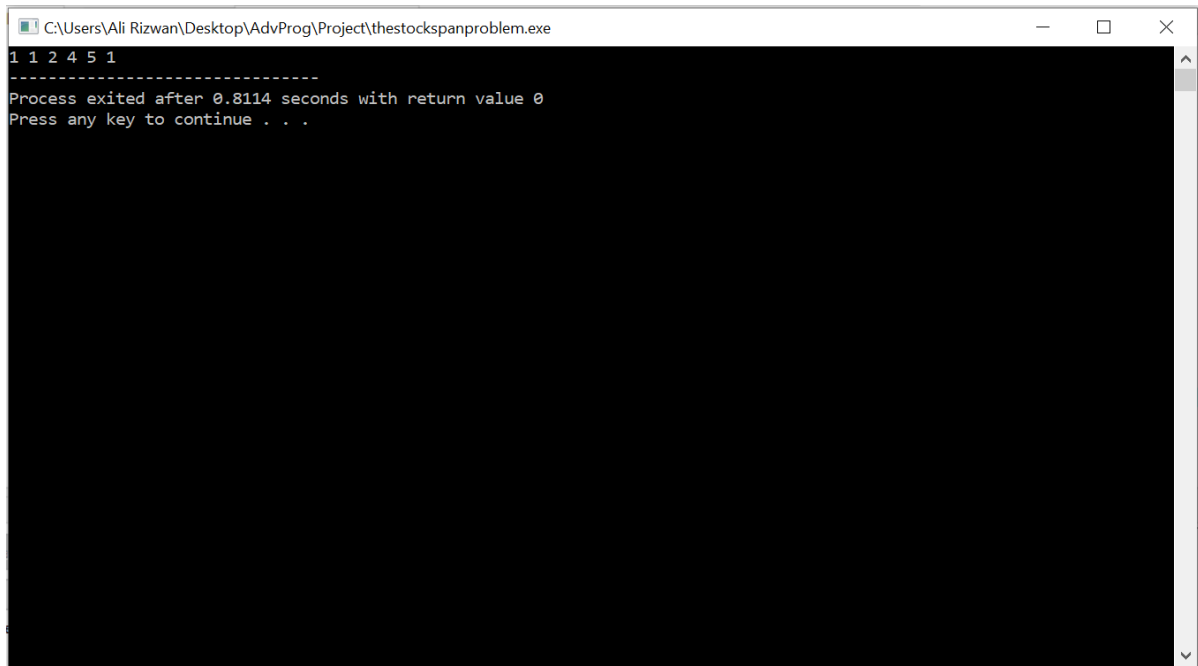
3.2 Coding:

```
1  #include <stdio.h>
2  int i,j;
3  void calculateSpan(int price[], int n, int S[])
4  {
5      S[0] = 1;
6
7      for ( i = 1; i < n; i++) {
8          S[i] = 1;
9
10         for ( j = i - 1; (j >= 0) && (price[i] >= price[j]); j--)
11             S[i]++;
12     }
13 }
14
15 void printArray(int arr[], int n)
16 {
17     for ( i = 0; i < n; i++)
18         printf("%d ", arr[i]);
19 }
20
21 int main()
22 {
23     int price[] = { 10, 4, 5, 90, 120, 80 };
24     int n = sizeof(price) / sizeof(price[0]);
25     int S[n];
26     calculateSpan(price, n, S);
27     printArray(S, n);
28
29     return 0;
30 }
31
```

Compile Log Debug Find Results

Fig 3.2.1

3.3Output:



```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\thestockspanproblem.exe
1 1 2 4 5 1
-----
Process exited after 0.8114 seconds with return value 0
Press any key to continue . . .
```

Fig 3.3.1

4.PROBLEM 3: A PRODUCT ARRAY PUZZLE

This Project aims to calculate the product of the given array.

4.1 Problem Statement:

Given an array `arr[]` of `n` integers, construct a Product Array `prod[]` (of same size) such that `prod[i]` is equal to the product of all the elements of `arr[]` except `arr[i]`.

Solve it without division operator in $O(n)$ time.

Concepts Used:

1. **Pointers:** A pointer is a variable that stores the address of another variable. Unlike other variables that hold values of a certain type, pointer holds the address of a variable.

Syntax: `int *ptr;`

2. **For loop:** A For loop is a control flow statement for specifying iteration, which allows code to be executed repeatedly.

Syntax: `for(initialization,condition,increment)`

```
{  
    Statements;  
}
```

3. **Arrays:** An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

Syntax: `int array[100];`

4. **Memory Allocation:** C Dynamic Memory Allocation can be defined as a procedure in which the size of a data structure (like Array) is changed during the runtime.

C provides some functions to achieve these tasks. There are 4 library functions provided by C defined under `<stdlib.h>` header file to facilitate dynamic memory allocation in C programming. They are:

- `malloc()`
- `calloc()`
- `free()`
- `realloc()`

5. If Statement: The if-else statement in C is used to perform the operations based on some specific condition. The operations specified in if block are executed if and only if the given condition is true.

```
Syntax: if {  
    Statements;  
}  
else{  
    Statements;  
}
```

Test Data:

Expected Output:

The product array is:

180 600 360 300 900

4.2 Coding:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void productArray(int arr[], int n)
4  {
5      if (n == 1) {
6          printf("0");
7          return;
8      }
9      int* left = (int*)malloc(
10         sizeof(int) * n);
11      int* right = (int*)malloc(
12         sizeof(int) * n);
13      int* prod = (int*)malloc(
14         sizeof(int) * n);
15      int i, j;
16      left[0] = 1;
17      right[n - 1] = 1;
18      for (i = 1; i < n; i++)
19          left[i] = arr[i - 1] * left[i - 1];
20      for (j = n - 2; j >= 0; j--)
21          right[j] = arr[j + 1] * right[j + 1];
22      for (i = 0; i < n; i++)
23          prod[i] = left[i] * right[i];
24      for (i = 0; i < n; i++)
25          printf("%d ", prod[i]);
26      return;
27 }
```

Fig 4.2.1

```
28 int main()
29 {
30     int arr[] = { 10, 3, 5, 6, 2 };
31     int n = sizeof(arr) / sizeof(arr[0]);
32     printf("The product array is: \n");
33     productArray(arr, n);
34     getchar();
35 }
36
```




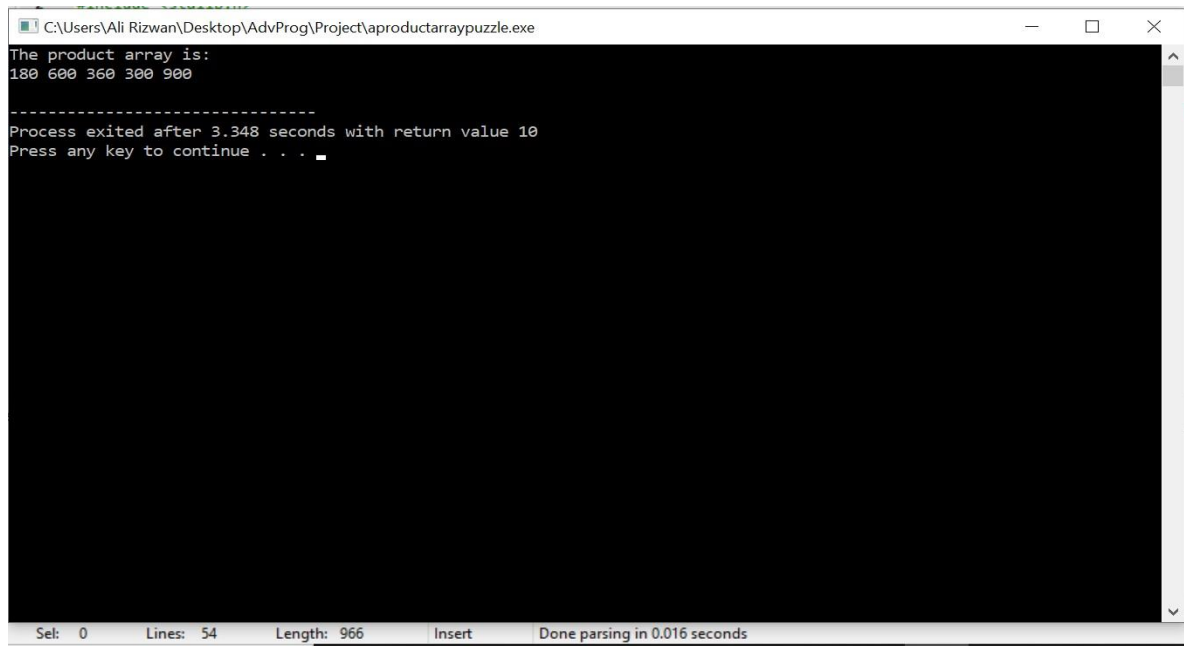
es  Compile Log  Debug  Find Results

Fig 4.2.2

4.3 Output:



```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\aproductarraypuzzle.exe
The product array is:
180 600 360 300 900

-----
Process exited after 3.348 seconds with return value 10
Press any key to continue . . .
```

Sel: 0 Lines: 54 Length: 966 Insert Done parsing in 0.016 seconds

Fig 4.3.1

5.PROBLEM 4: A PANCAKE SORTING PROBLEM

This project aims to Sort the elements in an array by Flipping the elements in an array up to a point and then Sorting the list of elements.

5.1 Problem Statement:

Given an unsorted array, sort the given array. You are allowed to do only following operation on array.

flip(arr,i):Reverse an array from 0 to i

Unlike a traditional sorting algorithm, which attempts to sort with the fewest comparisons possible, the goal is to sort the sequence in as few reversals as possible.

Concepts Used:

1. Arrays: An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

Syntax: int array[100];

2. For Loop: A For loop is a control flow statement for specifying iteration, which allows code to be executed repeatedly.

Syntax: for(initialization,condition,increment)

```
{  
    Statements;  
}
```

3. While Loop: A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

Syntax: while(Condition)

```
{  
    Statements;  
}
```

4. Recursion: Recursion is a programming technique that allows the programmer to express operations in terms of themselves. In C, this takes the form of a function that calls itself. A useful way to think of recursive functions is to imagine them as a process being performed where one of the instructions is to "repeat the process".

5. If Statement: The if-else statement in C is used to perform the operations based on some specific condition. The operations specified in if block are executed if and only if the given condition is true.

```
Syntax: if {  
    Statements;  
}  
Else{  
    Statements;  
}
```

6. Insertion Sort: Insertion sort algorithm picks elements one by one and places it to the right position where it belongs in the sorted list of elements.

Test Cases:

Expected Output :

6 12 18 20 30 35 35 40

80 90

5.2 Coding:

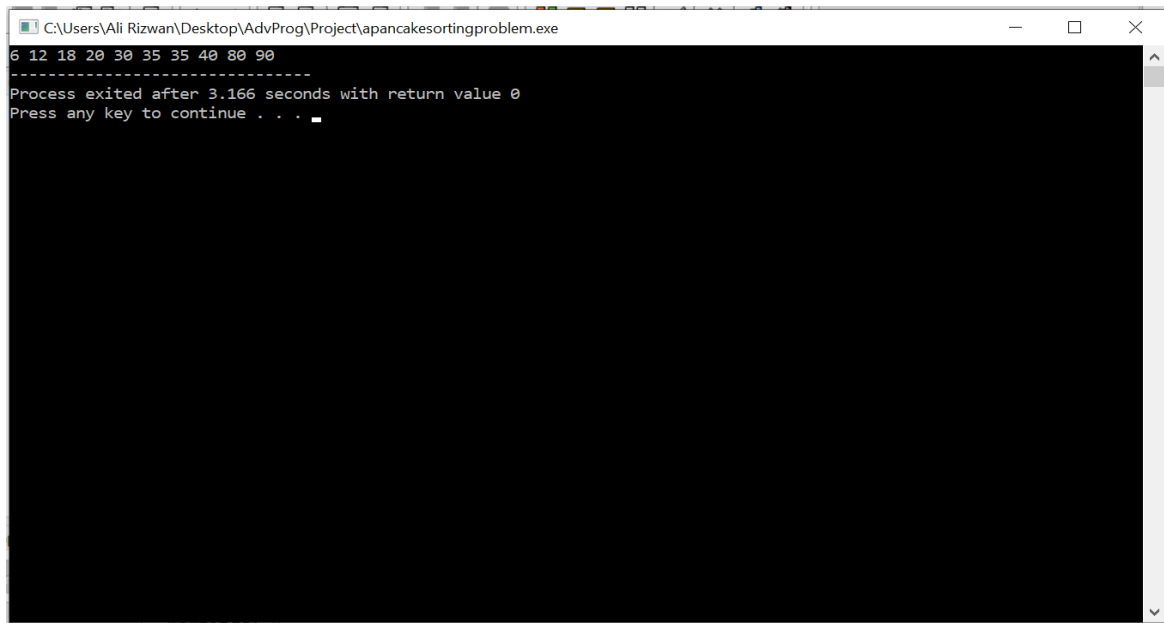
```
1  #include <stdlib.h>
2  #include <stdio.h>
3  int ceilSearch(int arr[], int low, int high, int x)
4  {
5      int mid;
6      if(x <= arr[low])
7          return low;
8      if(x > arr[high])
9          return -1;
10     mid = (low + high)/2;
11     if(arr[mid] == x)
12         return mid;
13     if(arr[mid] < x)
14     {
15         if(mid + 1 <= high && x <= arr[mid+1])
16             return mid + 1;
17         else
18             return ceilSearch(arr, mid+1, high, x);
19     }
20     if (mid - 1 >= low && x > arr[mid-1])
21         return mid;
22     else
23         return ceilSearch(arr, low, mid - 1, x);
24 }
25 void flip(int arr[], int i)
26 {
27     int temp, start = 0;
28     while (start < i)
29     {
30         temp = arr[start];
31         arr[start] = arr[i];
32         arr[i] = temp;
33         start++;
34     }
35 }
```

Fig 5.2.1

```
34     i--;
35 }
36 }
37 void insertionSort(int arr[], int size)
38 {
39     int i, j;
40     for(i = 1; i < size; i++)
41     {
42         int j = ceilSearch(arr, 0, i-1, arr[i]);
43         if (j != -1)
44         {
45             flip(arr, j-1);
46             flip(arr, i-1);
47             flip(arr, i);
48             flip(arr, j);
49         }
50     }
51 }
52 void printArray(int arr[], int n)
53 {
54     int i;
55     for (i = 0; i < n; ++i)
56         printf("%d ", arr[i]);
57 }
58 int main()
59 {
60     int arr[] = {18, 40, 35, 12, 30, 35, 20, 6, 90, 80};
61     int n = sizeof(arr)/sizeof(arr[0]);
62     insertionSort(arr, n);
63     printArray(arr, n);
64     return 0;
65 }
```

Fig 5.2.2

5.3Output:



```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\apancakesortingproblem.exe
6 12 18 20 30 35 35 40 80 90
-----
Process exited after 3.166 seconds with return value 0
Press any key to continue . . .
```

Fig 5.3.1

6.PROBLEM 5: BILL PAYMENT RECEIPT

This problem enables us to generate the Bill Receipts in the Super market.

6.1Problem Statement:

In this problem ,we have to take the input from the user such as the item code, item name, quantity of the product and Price of the product .The Output Should be Printed with total number of items that are brought and the total Amount to be paid by the Customer.

Concepts Used:

1. While loop: A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

```
Syntax: while{  
        Statements;  
    }
```

2. For Loop: A For loop is a control flow statement for specifying iteration,which allows code to be executed repeatedly.

```
Syntax: for(initialization,condition,increment)  
    {  
        Statements;  
    }
```

3. Arrays: An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

```
Syntax: int array[100];
```

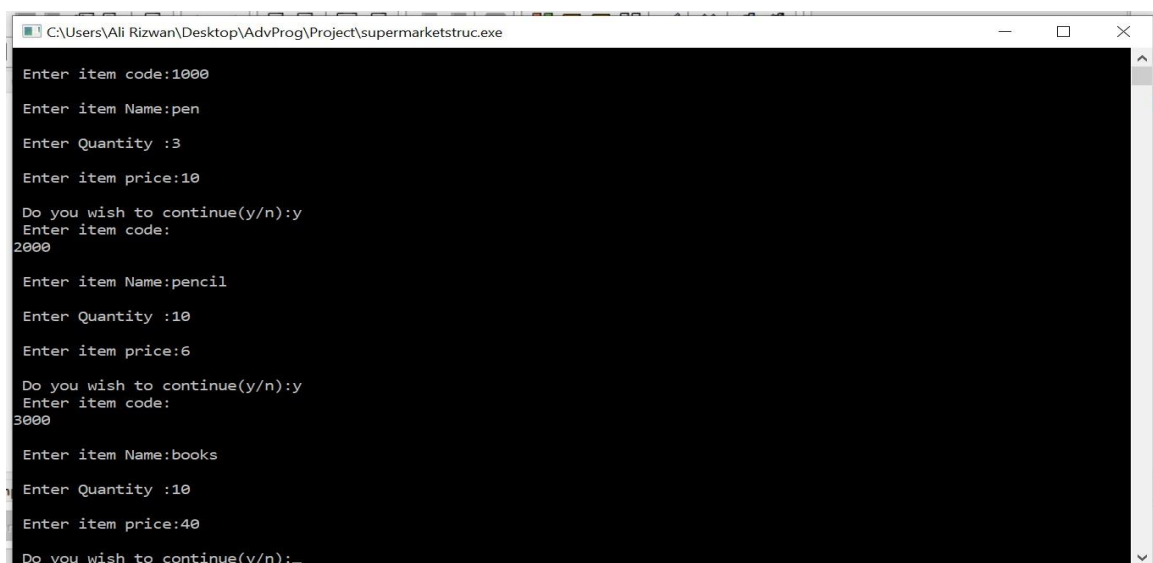
4. Structures: Arrays allow to define type of variables that can hold several data items of the same kind. Similarly structure is another user defined data type available in C that allows to combine data items of different kinds.

5. Logical Operator OR (||): The logical-OR operator performs an inclusive-OR operation on its operands. The result is 0 if both operands have 0 values. If either operand has a nonzero value, the result is 1. If the first operand of a logical-OR operation has a nonzero value, the second operand is not evaluated.

Syntax: if(a || b)
{
Statements;
}

Test Cases :

Expected Output :



```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\supermarketstruc.exe
Enter item code:1000
Enter item Name:pen
Enter Quantity :3
Enter item price:10
Do you wish to continue(y/n):y
Enter item code:
2000
Enter item Name:pencil
Enter Quantity :10
Enter item price:6
Do you wish to continue(y/n):y
Enter item code:
3000
Enter item Name:books
Enter Quantity :10
Enter item price:40
Do you wish to continue(y/n):
```

Fig 6.1.1


```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\supermarketstruc.exe

Enter item Name:pencil
Enter Quantity :10
Enter item price:6
Do you wish to continue(y/n):y
Enter item code:
3000

Enter item Name:books
Enter Quantity :10
Enter item price:40
Do you wish to continue(y/n):n
-----Vijetha Super Market-----
Sno      Icode   Name    Price   Qty     Amount
=====
1         1000    pen     10.000000    3     30.000000
2         2000    pencil  6.000000    10     60.000000
3         3000    books   40.000000    10    400.000000
No. of Items:3
Total Bill Amount:490.000000
-----
Process exited after 84.8 seconds with return value 30
Press any key to continue . . .
- warnings: 0
```

Fig 6.1.2

6.2 Coding:

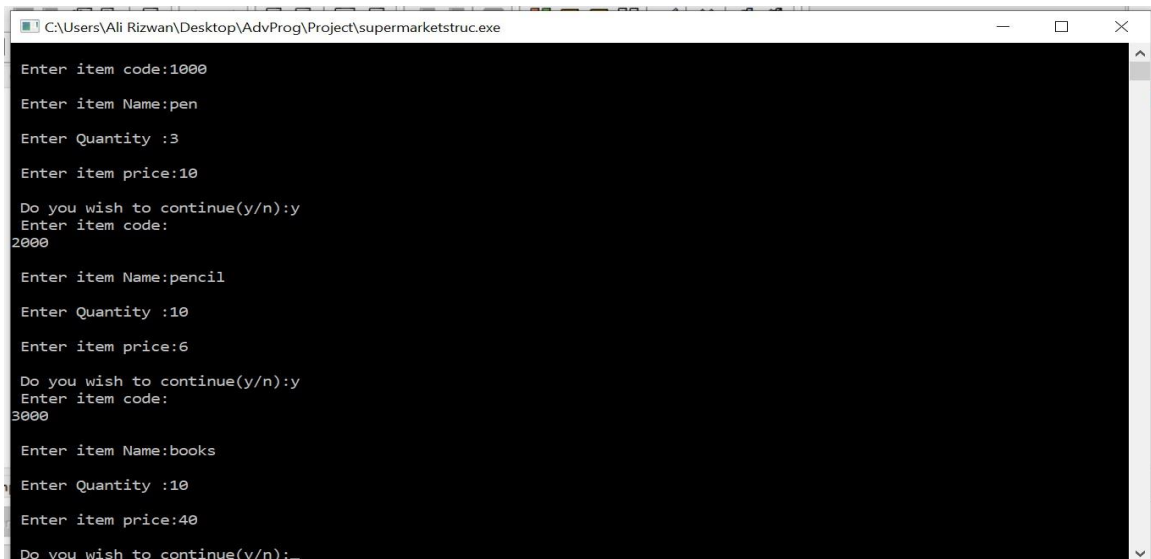
```
1  #include<stdio.h>
2  main()
3  {
4      struct supermarket
5      {
6          int icode,qty;
7          char iname[20];
8          float price,amount;
9      }s[100];
10     int i=0,count=0,sno;
11     float tamount=0;
12     char ch='y';
13     while(ch=='y' || ch=='Y')
14     {
15         printf("\n Enter item code:");
16         scanf("%d",&s[i].icode);
17         printf("\n Enter item Name:");
18         scanf("%s",&s[i].iname);
19         printf("\n Enter Quantity :");
20         scanf("%d",&s[i].qty);
21         printf("\n Enter item price:");
22         scanf("%f",&s[i].price);
23         s[i].amount=s[i].qty*s[i].price;
24         tamount=tamount+s[i].amount;
25         i++;
26         count++;
27         printf("\n Do you wish to continue(y/n):");
28         ch=getche();
29     }
```

Fig 6.2.1

```
30     printf("\n----Vijetha Super Market-----");
31     printf("\n Sno\t Icode\t Name \t Price \t Qty \t Amount");
32     printf("\n=====");
33     for(i=0,sno=1;i<count;i++,sno++)
34     {
35         printf("\n%d \t%d \t%s \t%f \t%d \t%f",sno,s[i].icode,s[i].iname,s[i].price,s[i].qty,s[i].amount);
36     }
37     printf("\n No. of Items:%d",count);
38     printf("\n Total Bill Amount:%f",tamount);
39 }
```

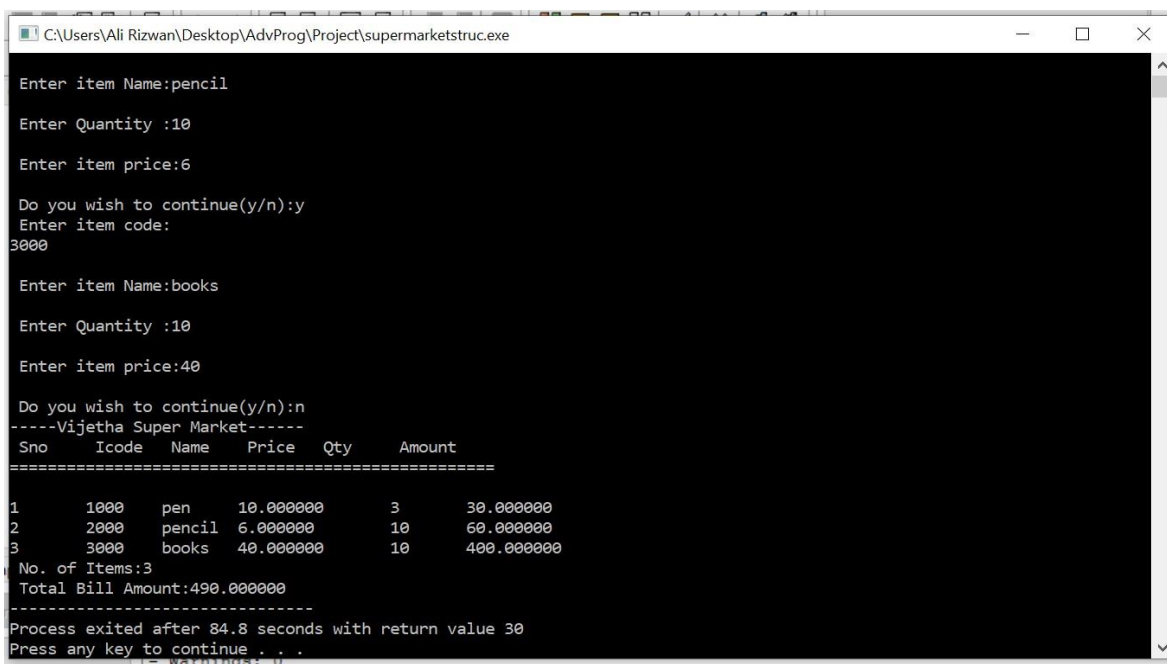
Fig 6.2.2

6.3Output:



```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\supermarketstruc.exe
Enter item code:1000
Enter item Name:pen
Enter Quantity :3
Enter item price:10
Do you wish to continue(y/n):y
Enter item code:
2000
Enter item Name:pencil
Enter Quantity :10
Enter item price:6
Do you wish to continue(y/n):y
Enter item code:
3000
Enter item Name:books
Enter Quantity :10
Enter item price:40
Do you wish to continue(y/n):
```

Fig 6.3.1



```
C:\Users\Ali Rizwan\Desktop\AdvProg\Project\supermarketstruc.exe
Enter item Name:pencil
Enter Quantity :10
Enter item price:6
Do you wish to continue(y/n):y
Enter item code:
3000
Enter item Name:books
Enter Quantity :10
Enter item price:40
Do you wish to continue(y/n):n
-----Vijetha Super Market-----
Sno      Icode  Name    Price  Qty    Amount
=====
1         1000   pen     10.000000    3    30.000000
2         2000   pencil  6.000000    10   60.000000
3         3000   books   40.000000    10  400.000000
No. of Items:3
Total Bill Amount:490.000000
-----
Process exited after 84.8 seconds with return value 30
Press any key to continue . . .
```

Fig 6.3.2

7. SOFTWARE REQUIREMENTS

7.1 Hardware Requirements:

This project can be executed in any system or an android phone without prior to any platform.

We can use any online compiler and interpreter.

7.2 Software Requirements:

There are two ways to execute this project

1. Online compilers
2. Softwares for execution (DEV C++,TURBO C,...)

Online Compilers require only internet connection. We have many free compilers with which we can code.

Softwares for execution need to be installed based on the user's system specification. These help us to completely execute the project. These softwares are based on the platforms

REFERENCES

1. https://en.wikipedia.org/wiki/Reservoir_sampling
2. <https://www.includehelp.com/icp/stock-span-problem.aspx>
3. <https://www.tutorialspoint.com/a-product-array-puzzle-in-cplusplus#:~:text=There%20is%20an%20array%20with,division%20operator%20in%20this%20problem.>
4. <https://www.tutorialcup.com/interview/array/pancake-sorting-problem.htm>

