Advanced Programming

M S S BHARADWAJ

221710313031

Guide: Ms. B. K. V. P. S. Mahalakshmi

Contents:

1. Concepts Learned

During my internship I have learned following :

- Basics of C
- Data structures with C
- Basics of C++
- Basics of Python
- Python (Advanced)
- Problem Solving (Coding)

Basics Of C

C is a procedural programming language. It was initially developed by Dennis Ritchie as a system programming language to write operating system. The main features of C language include low-level access to memory, simple set of keywords, and clean style, these features make C language suitable for system programming like operating system or compiler development.

- Variable Declaration
- Input/Output
- Data types
- Conditionals
- Loops
- Arrays
- Strings
- Pointers
- Functions
- Structures
- Unions
- Files

Data Structures With C

Data Structures in C are used to store data in an organised and efficient manner. The C Programming language has many data structures like **stack**, **queue**, **linked list**, **tree**, etc. A programmer selects an appropriate data structure and uses it according to their convenience.

- Linked List
- Stack
- Queue
- Trees
- Graph

Basics Of C++

C++ is a general purpose programming language and widely used now a days for competitive programming. It has imperative, object-oriented and generic programming features. C++ runs on lots of platform like Windows, Linux, Unix, Mac etc.

- Program structure
- Input/Output
- Operators
- Functions
- Classes
- Constructor and Destructor
- Function Overloading
- Operator Overloading
- Inheritance
- Polymorphism

Basics Of Python

Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written.

- Variable Declaration
- Input/Output
- Data types
- Conditionals
- Loops
- Lists
- Tuple
- Dictionary
- Sets
- Methods
- Packages

Python (Advanced)

Python is not just for simple programming. It is used for machine learning, data refinement, data analytics and many more. It is the language which solves many of the industrial problem very easily

- Python Pandas
- Python Numpy
- Python Data frames
- Python Visualization
- Python for Data Science
- Python for Machine Learning

Problem Solving (Coding)

Developers and students solve a lot of coding questions of data structures and algorithms but most of them don’t understand the importance of it. A lot of them also have this opinion that data structure and algorithms only help in interviews and after that, there is no use of all those complicated stuff in daily jobs. In real-world projects, your brain should be able to write a quick efficient solution for complicated stuff and you can only do that when you practice a lot of coding questions. Understand that language and frameworks are just tools, it won’t teach you problem-solving skills. You develop problem-solving skills when you practice a lot of coding questions.

2. Works Carried Out

Works carried out by me during my internship are:

- Python Programming
- Python (Data science and Machine Learning)
- Programming with C
- Object Oriented Programming with C++
- Solving Coding Challenges

Python Programming

I have done following works under Python programming:

- Solved 50+ Basics Level Problems using Python
- Solved 25+ Coding Competition Problems
- Submitted More than 25+ assignments given by the company

Python (Data Science)

I have done following works under Python :

- Learnt using python for data preprocessing
- Solved 10+ using Python problems related to data manipulation
- Performed visualizations with python
- Submitted More than 10+ assignments given by the company

Problem Solving With C

I have done following works under C programming:

- Solved 100+ Basics Level Problems using C
- Solved 25+ Coding Competition Problems
- Submitted More than 25+ assignments given by the company

Object Oriented Programming With C++

I have done following works under C++ programming:

- Solved 20+ Basics Level Problems using C++
- Solved 5+ Coding Competition Problems
- Submitted More than 3+ assignments given by the company

Solving Coding Challenges

I have done following works under Python programming:

- Solved 20+ Coding Competition Problems
- Submitted More than 5+ assignments given by the company

3. Internship - Learning Methodology

During my internship we have followed a step by step methodology to achieve our goal. The steps of followed methodology are:

1. Learning concepts
2. Solving basic coding problems
3. Solving advanced coding problems
4. Solving coding competition problems

Learning Concepts

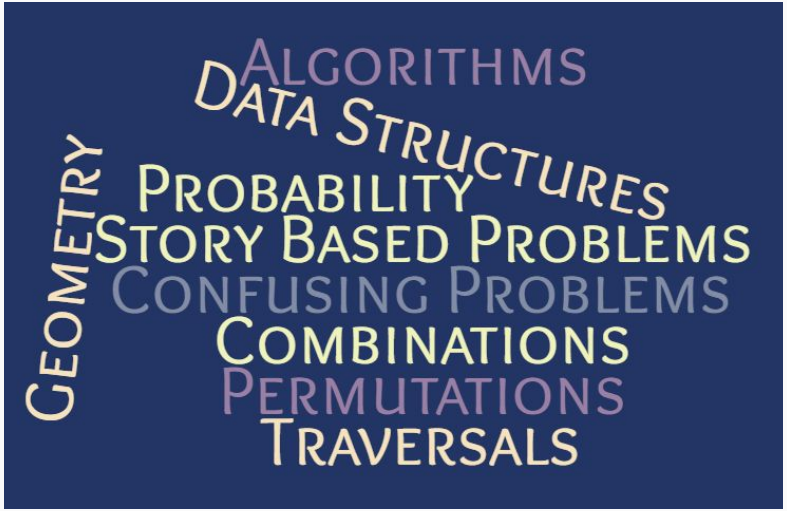
Learning concepts is one basic step of learning as it builds a strong foundation about the subject. Which helps us to apply that particular subject in any situation. Learning concepts is a step which can be described as pillars of a building.

Solving Basic Coding Problems

After learning concepts, application of those concepts is done to understand basics. In this step we will solve very basic level coding to learn how to use and what to use.

Solving Advanced Coding Problems

After the basic level of coding we will solve advanced coding. These problems need more logical thinking and question understanding . These type of questions are mostly asked in technical rounds of hiring.



Solving Coding Competition Problems

After practising advanced level problems, getting prepared for next level is important. The next level is to practise coding competition question and participating in them. The questions in these will be a mix of basics concepts with advanced one.

4. Details Of Project / Problems Carried Out

There are eight problems which are listed below are of different complexity and require different approach and logics in order to achieve the desired Output/ Solution.

1. **Last digit** - In this problem, we compute the last digit of a sequence which is calculated with the given formula.
2. **Replace occurrences with the given words -** In this problem, we search for a particular word from the input and we replace it with another word.
3. **Bike tour -** In this problem, we calculate the number of peak points as per given conditions.
4. **Robot path decoding -** In this problem, we calculate the position of the robot based on the given input.
5. **Perfect subarray** - In this problem, we find out the count of subarrays which are perfect as per given conditions.
6. **Big city skylines** - In this problem, we calculate the maximum area rectangle from inputs.
7. **ATM** - In this problem, we check multiple transactions based on the given conditions.
8. **Cryptopangrams** - In this problem, we recover pangrams based on the inputs.

CRYPTOPANGRAMS

```
def gcd(a, b):
    while b != 0:
        a, b = b, a%b
    return a

T = int(input())
for t in range(T):
    N, L = map(int, input().split())
    a = list(map(int, input().split()))
    p = [0]*(L+1)
    for i in range(L-1):
        c = gcd(a[i], a[i+1])
        if c != a[i] and c != a[i+1]:
            p[i+1] = c
    for i in range(L):
        if p[i] != 0 and p[i+1] == 0:
            p[i+1] = a[i] // p[i]
    for i in range(L, 0, -1):
        if p[i] != 0 and p[i-1] == 0:
            p[i-1] = a[i-1] // p[i]
    b = sorted(set(p))
    d = dict()
    for i in range(26):
        d[b[i]] = chr(ord('A') + i)
    s = ""
    for i in p:
        s += d[i]
    print("Case #" + str(t+1) + ": " + s)
```

ATM

```
n=input().split(' ');

k=float(n[1])
n1=int(n[0])
if(n1%5==0 and n1<=k-0.5):
    n1=float(n1)+0.5;
    k=k-n1;
    print("%.2f" % k)
else:
    print("%.2f" % k)
```

PERFECT SUBARRAY

```
import math
t=int(input())
for i in range(1,t+1):
    n=int(input())
    list1=list(map(int,input().split()))
    sublist = [math.sqrt(sum(list1[k:j]))-math.floor(math.sqrt(sum(list1[k:j])))==0
               for k in range(len(list1) + 1) for j in range(k + 1, len(list1) + 1)]
    print("Case #"+str(i)+": "+str(sublist.count(1)))
```

BIG CITY SKYLINES

```
n=int(input())
a=list(map(int,input().split()))
a1=[a[i] for i in range(0,len(a)) if i%2==0]
a2=[a[i] for i in range(0,len(a)) if i%2!=0]
print(sum(a1)*min(a2))
```

ROBOT PATH DECODING

```
t=int(input())
for t in range(1, t+1):
    p=input()
    cur=[0, 0]
    stack=[]
    for char in p:
        if char == 'N':
            cur[0] -= 1
        elif char == 'S':
            cur[0] += 1
        elif char == 'W':
            cur[1] -= 1
        elif char == 'E':
            cur[1] += 1
        elif char.isdigit():
            stack.append((cur[0], cur[1], int(char)))
        elif char == '(':
            cur = [0, 0]
        elif char == ')':
            pop = stack.pop()
            cur = [pop[i] + pop[2]*cur[i] for i in [0, 1]]
    final_row = (1+cur[0])%10**9
    if final_row == 0:
        final_row = 10**9

    final_column = (1+cur[1])%10**9
    if final_column == 0:
        final_column = 10**9
    print("Case #%d: %d %d" % (t, final_column, final_row))
```

BIKE TOUR

```
t=int(input())
for i in range(1,t+1):
    n=int(input())
    a=list(map(int,input().split()))
    c=0
    for j in range(1,n-1):
        if (a[j-1]<a[j] and a[j+1]<a[j]):
            c+=1
    print("Case #"+str(i)+": "+str(c))
```

REPLACE OCCURENCES WITH THE GIVEN WORD

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
    char string[1010],a[30],b[30];
    scanf ("%s", a);
    scanf ("%s", b);
    fflush (stdin);
    scanf ("%[^\n]s", string);
    char *p;
    p = strtok (string, " ");
    while (p != NULL)
    {
        if (strcmp (a, p) == 0)
        printf ("%s ", b);
        else
        printf ("%s ", p);
        p = strtok (NULL, " ");
    }
return 0;
}
```

LAST DIGIT

```
#include<stdio.h>
#include<math.h>
int main()
{
 int i,j,n,b,s=0,a;
 scanf("%d",&n);
 for(i=0;pow(2,i)<=n;i++)
 {
    a=pow(2,i);
    for(j=0;j<=n;j++)
    {
        b=pow(2,(a+2*j));
        s=s+b;
    }
 }
 printf("%d",s%10);
 return 0;
}
```

CRYPTOPANGRAMS

```
2
103 31
217 1891 4819 2291 2987 3811 1739 2491 4717 445 65 1079 8383 5353 901 187 649 1003 697 3239 7663 291 123 779 1007 3551 1943 211
7 1679 989 3053
Case #1: CJQUIZKNOWBEVYOFDPFLUXALGORITHMS
10000 25
3292937 175597 18779 50429 375469 1651121 2102 3722 2376497 611683 489059 2328901 3150061 829981 421301 76409 38477 291931 7302
41 959821 1664197 3057407 4267589 4729181 5335543
Case #2: SUBDERMATOGLYPHICFJKNQVWXZ
```

ATM

```
30 120
89.50
```

BIG CITY SKYLINES

```
5
5 7 1 20 3 5 6 3 2 10
51
```

PERFECT SUBARRAY

```
3
3
2 2 6
Case #1: 1
5
30 30 9 1 30
Case #2: 3
4
4 0 0 16
Case #3: 9
```

BIKE TOUR

```
4
3
10 20 10
Case #1: 1
4
7 7 7 7
Case #2: 0
5
10 90 20 90 10
Case #3: 2
```

ROBOT PATH DECODING

```
4
SSSEEE
Case #1: 4 4
N
Case #2: 1 1000000000
N3(S)N2(E)N
Case #3: 3 1
2(3(NW)2(W2(EE)W))
Case #4: 3 999999995
```

REPLACE OCCURENCES WITH THE GIVEN WORD

```
tiger lion
the tiger is a wild animal the tiger is known as the king of the jungle
the lion is a wild animal the lion is known as the king of the jungle
```

LAST DIGIT

```
3
0
```

Any Questions

Thank You