[Thoughts of Praveen](http://programmingandaptitude.blogspot.in/)

* [Home](http://programmingandaptitude.blogspot.nl/2015/07/home.html)
* [Aptitude](http://programmingandaptitude.blogspot.nl/2015/06/aptitude-basics.html)
* [C Programming](http://programmingandaptitude.blogspot.nl/2015/07/zoho-interview-questions-and-answer.html)
* [Interview](http://programmingandaptitude.blogspot.nl/2015/06/basic-networking-interview-questions.html)

**MONDAY, 13 JULY 2015**

zoho interview questions and answer

[**ZOHO**](https://www.zoho.com/)**INTERVIEW QUESTIONS**

[](http://3.bp.blogspot.com/-bGe0xboVOB0/VaR__7l-ZlI/AAAAAAAAAQ8/JQvQ5jQ8fGA/s1600/Zoho_logo.png)

**ROUND 1: WRITTEN**

APTITUDE(1 hr and 20 minutes-20 questions): Problems on average, probability, time & distance, alligation&mixture,ratio, HCF & LCM and few a puzzles.

TECHNICAL(45 minutes-10 questions): Output for C questions. Practice questions in geekquiz.com and C output questions in geeksforgeeks.org. Questions in pointers, strings, matrix etc.

Nearly 60 students were selected out of 600 candidates. They didn’t select the top 60. They had a cutoff and those who cleared the cutoff were called for the next round

**ROUND 2: SIMPLE CODING(3 hours)**

1. Write a program to give the following output for the given input

Eg 1: Input: a1b10

Output: abbbbbbbbbb

Eg: 2: Input: b3c6d15

Output: bbbccccccddddddddddddddd

The number varies from 1 to 99.

2. Write a program to sort the elements in odd positions in descending order and elements in ascending order

Eg 1: Input: 13,2 4,15,12,10,5

Output: 13,2,12,10,5,15,4

Eg 2: Input: 1,2,3,4,5,6,7,8,9

Output: 9,2,7,4,5,6,3,8,1

3. Write a program to print the following output for the given input. You can assume the string is of odd length

Eg 1: Input: 12345

Output:

1 5

2 4

3

2 4

1 5

Eg 2: Input: geeksforgeeks

Output:

g s

e k

e e

k e

s g

f r

o

f r

s g

k e

e e

e k

g s

4. Find if a String2 is substring of String1. If it is, return the index of the first occurrence. else return -1.

Eg 1:Input:

String 1: test123string

String 2: 123

Output: 4

Eg 2: Input:

String 1: testing12

String 2: 1234

Output: -1

5. Given two sorted arrays, merge them such that the elements are not repeated

Eg 1: Input:

Array 1: 2,4,5,6,7,9,10,13

Array 2: 2,3,4,5,6,7,8,9,11,15

Output:

Merged array: 2,3,4,5,6,7,8,9,10,11,13,15

6. Using Recursion reverse the string such as

Eg 1: Input: one two three

Output: three two one

Eg 2: Input: I love india

Output: india love I

**ROUND 3: COMPLEX CODING(3 hours)**

1) Design a Call taxi booking application  
-There are n number of taxi’s. For simplicity, assume 4. But it should work for any number of taxi’s.  
-The are 6 points(A,B,C,D,E,F)  
-All the points are in a straight line, and each point is 15kms away from the adjacent points.  
-It takes 60 mins to travel from one point to another  
-Each taxi charges Rs.100 minimum for the first 5 kilometers and Rs.10 for the subsequent kilometers.  
-For simplicity, time can be entered as absolute time. Eg: 9hrs, 15hrs etc.  
-All taxi’s are initially stationed at A.  
-When a customer books a Taxi, a free taxi at that point is allocated  
-If no free taxi is available at that point, a free taxi at the nearest point is allocated.  
-If two taxi’s are free at the same point, one with lower earning is allocated  
-Note that the taxi only charges the customer from the pickup point to the drop point. Not the distance it travels from an adjacent point to pickup the customer.  
-If no taxi is free at that time, booking is rejected

Design modules for

1) Call taxi booking

Input 1:

Customer ID: 1

Pickup Point: A

Drop Point: B

Pickup Time: 9

Output 1:

Taxi can be allotted.

Taxi-1 is allotted

Input 2:

Customer ID: 2

Pickup Point: B

Drop Point: D

Pickup Time: 9

Output 1:

Taxi can be allotted.

Taxi-2 is allotted

(Note: Since Taxi-1 would have completed its journey when second booking is done, so Taxi-2 from nearest point A which is free is allocated)

Input 3:

Customer ID: 3

Pickup Point: B

Drop Point: C

Pickup Time: 12

Output 1:

Taxi can be allotted.

Taxi-1 is allotted

2) Display the Taxi details

Taxi No: Total Earnings:

BookingID CustomerID From To PickupTime DropTime Amount

Output:

Taxi-1 Total Earnings: Rs. 400

1 1 A B 9 10 200

3 3 B C 12 13 200

Taxi-2 Total Earnings: Rs. 350

2 2 B D 9 11 350

These were just sample inputs. It should work for any input that they give.  
Those who finished both the modules within 3 hours and if it worked for all the inputs they give, those candidates were given extra modules to work with.

**ROUND 4 : FIRST FACE-TO-FACE(TECHNICAL)**

Questions were on project, c, oops concepts, DBMS and a few puzzles. They might ask you more on new scenarios relating to your project.

**ROUND 5: SECOND FACE-TO-FACE(TECHNICAL)**

Question were on c, c++, java(like threads, synchronization etc.), Discussion about questions from first, second and third round. He even asked me to solve a few questions from the first round. He gave me a few puzzles to solve

**ROUND 6: FIRST GENERAL HR**

General questions about my pros and cons and discussion on my resume(be thorough with your resume). She finally asked me if I had any queries.

**ROUND 7: SECOND GENERAL HR**

She asked me some family details and gave some scenarios and asked me to what I will do in such situations(like if I am given the power to change 3 things in india, what all will I change) and a few general questions.

**ANSWER;**

**ROUND 2;**

1. **A10b2**

#include<stdio.h>

#include<string.h>

int main()

{

char a[100],t;

int i,j,x,f;

printf("Enter the string:");

scanf("%s",&a);

for(i=0;i<strlen(a);i=i+2)

{

t=a[i];

if((a[i+2]-48)>=0 && (a[i+2]-48)<=9)

{

int x;

x=((a[i+1]-48)\*10)+(a[i+2]-48);

for(f=0;f<x;f++)

{

printf("%c",t);

}

i++;

}

else

{

for(j=0;j<a[i+1]-48;j++)

{

printf("%c",t);

}

}

}

return 0;

}

Reverse process

Input:aabba

Output:a3b2

Java: public class JavaApplication3 {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

String str=s.next();

Map<Character,Integer> map = new HashMap<>();

for (int i = 0; i < str.length(); i++) {

char c = str.charAt(i);

if (map.containsKey(c)) {

int cnt = map.get(c);

map.put(c, ++cnt);

} else {

map.put(c, 1);

}

}

System.out.println(map);

}

}

**Output**

aaabbbdj

{a=3, b=3, d=1, j=1}

**2)SORTING WITH ODD-DESCENDING AND EVEN-ASCENDING**

#include<iostream.h>

#include<conio.h>

int main()

{

int n,i,j,temp,\*a=NULL;

cout<<"enetr the size:";

cin>>n;

a=new int[n];

for(i=0;i<n;i++)

{

cin>>a[i];

}

for(i=0;i<n;i=i+2)

{

for(j=i+2;j<n;j=j+2)

{

if(a[i]<a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

for(i=0;i<n;i++)

{

cout<<a[i];

}

getch();

return 200;

}

**4) SUBSTRING**

public class JavaApplication1 {  
   static int check(String str1,String str2)  
    {  
        String str;  
        int a=-1;  
        for(int i=0;i<=str1.length()-str2.length();i++)  
        {  
            str=str1.substring(i,i+str2.length());  
            if(str.equals(str2))  
                return i+1;  
        }  
        return a;  
    }  
    public static void main(String[] args) {  
        Scanner s=new Scanner(System.in);  
        String str1=s.next();  
        String str2=s.next();  
        System.out.println(check(str1,str2));  
          
    }  
      
}

**5)**

#include<conio.h>

#include<string.h>

void main()

{

int a[100];

int i,j,s,temp;

clrscr();

printf("Enter the size of array:");

scanf("%d",&s);

for(i=0;i<s;i++)

{

scanf("%d",&a[i]);

}

for(i=0;i<s;i+=2)

{

for(j=i+2;j<s;j+=2)

{

if(a[i]<a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

for(i=1;i<s;i+=2)

{

for(j=i+2;j<s;j+=2)

{

if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("array");

for(i=0;i<s;i++)

{

printf(" \t%d",a[i]);

}

getch();

}

**3) CROSS PATTERN**

#include<iostream.h>

#include<conio.h>

void main(){

clrscr();

int n;

cout<<"\n enter size:";

cin>>n;

char \*a=new char [n];

cout<<"\n Enter name:";

cin>>a;

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

if((i==j)||(i+j==n-1))

cout<<a[j];

cout<<"\t";

}

cout<<"\n";

}

getch();

}

**6) REVERSE THE SENTENCE**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

char a[1000];

int i,j,count=0;

clrscr();

gets(a);

for(i=strlen(a)-1;i>=0;i--)

{

count=count+1;

if(a[i]==' '|| i==0)

{

for(j=i;j<=i+count;j++)

{

printf("%c",a[j]);

}

count=0;

}

}

getch();

}

**Round 3: TAXI**

#include<stdio.h>  
#include<conio.h>  
#include<math.h>  
#define pf printf  
#define c() clrscr()  
  
int check2(char,int);  
int check\_picpnt(char,int);  
int booking();  
void details();  
int static id;  
char dist[6]={'A','B','C','D','E','F'};           /\* pick up points \*/  
struct taxi{  
     int index;                              /\*for each customer\*/  
     char pic\_pnt[10];  
     char drp\_pnt[10],cur\_pos;                   /\* struct variables \*/  
     int str\_time[10],cust\_id[10],end\_time[10];  
     int wage[10],t\_wage;  
}t[5];  
  
void main()                                      /\* main function \*/  
{  
  c();  
  for(int i=0;i<5;i++)  
  {                                              /\* initial position allot \*/  
  t[i].cur\_pos='A';  
  }  
  menu:  
  pf("\t\t\t\tWELCOME TO TAXI BOOKING");  
  pf("\n\n\n\n\n\t\t MENU");                    /\* front page \*/  
  pf("\n\n\n\t\t\t1.TAXI BOOKING");  
  pf("\n\t\t\t2.DETAILS\n");  
  pf("\t\t\t3.EXIT\n");  
  int opt;  
  pf("\n\n\t\t\tEnter your option:");  
  scanf("%d",&opt);  
  switch(opt)  
  {  
   case 1:  
   booking();                               /\* calling respective functions \*/  
   c();  
   goto menu;  
   case 2:  
   details();  
   getch();  
   c();  
   goto menu;  
   case 3:  
   break;  
   default:  
   pf("Wrong option!!!");  
   getch();  
   c();  
   goto menu;  
  }  
  getch();  
}  
  
int booking()  
{                                             /\*booking function,second page\*/  
c();  
char p\_pnt,d\_pnt;  
int tme,wag,taxi=0;  
pf("\t\t\tbooking menu");  
pf("\n\ncustomer\_id %d",++id);  
pf("\n\npickup\_pnt:");  
getchar();  
scanf("%c",&p\_pnt);                          /\* getting booking details from user using temp variables \*/  
pf("\ndrop\_pnt:");  
getchar();  
scanf("%c",&d\_pnt);  
pf("\ntime:");  
getchar();  
scanf("%d",&tme);  
taxi=check\_picpnt(p\_pnt,tme);            /\* checking taxi availability by calling function \*/  
if(taxi!=777)  
{  
int x=t[taxi].index;  
pf("\n\ntaxi %d is alloted",taxi+1);      /\* Assigning details to struct variables if function return respective taxis \*/  
pf("\n\nthank for booking");                   /\* display alloted taxi number \*/  
int cost=((abs(p\_pnt-d\_pnt)\*15)-5)\*10+100; /\* calculating cost based on distance\*/  
t[taxi].pic\_pnt[x]=p\_pnt;  
t[taxi].drp\_pnt[x]=d\_pnt;  
t[taxi].cust\_id[x]=id;  
t[taxi].wage[x]=cost;  
t[taxi].cur\_pos=d\_pnt;  
t[taxi].t\_wage=t[taxi].t\_wage+cost;  
t[taxi].end\_time[x]=abs(p\_pnt-d\_pnt)+tme; /\*calculating taxi free time \*/  
t[taxi].str\_time[x]=tme;  
t[taxi].index++;                    /\*increasing for each customer\*/  
}  
else  
{pf("\n\nOOPs failed");}            /\* taxi is not available\*/  
getch();  
return 0;  
}  
  
  
int check\_picpnt(char pic\_pnt,int p\_tme) /\*function to check the pick point\*/  
{  
int flag=0,min=10000,min\_ind,get;      /\* set the value for min\*/  
for(int i=0;i<5;i++)  
{  
 if(pic\_pnt==t[i].cur\_pos)  /\*check the pickpoint of the customer with the current position of the taxi\*/  
 {  
   if(t[i].end\_time[(t[i].index-1)]<=p\_tme)     /\* check the previously allocated time of the customer with the current customer\*/  
     {  
          if(t[i].t\_wage<min) /\* calculate the minimium wage who earned that day \*/  
           {  
            min=t[i].t\_wage;  
            min\_ind=i;/\* assigning the minimum index\*/  
            flag=1;  
           }  
     }  
 }  
}  
  if(flag==1)  
  {return (min\_ind);}  /\*returning the respective taxi index to the booking function \*/  
  
  if(flag==0) /\*if the taxi is not in that current position\*/  
  {  
    for(int j=0;j<6;j++) /\*getting the index of the pick point  \*/  
      {  
       if(pic\_pnt==dist[j])  
       {get=j;  
       }  
      }  
     int k,m,m1,indx,indx1;  
  for(j=get-1,k=get+1;j>=0||k<=6;j--,k++) /\* checking both the sides of the pickpoint if the taxi is available or not\*/  
    {  
      if(j>=0)  
      {  
          indx=check2(dist[j],p\_tme);/\* passing the neighbour value and pick up time \*/  
            if(indx!=-1)  
            m=t[indx].t\_wage; /\* storing the minimum wage in that current point \*/  
            else  
            m=10000;  
      }  
     if(k<=6)  
      {  
       indx1=check2(dist[k],p\_tme);  /\* passing the neighbour value and pick up time \*/  
           if(indx1!=-1)  
           m1=t[indx1].t\_wage;  /\* storing the minimum wage in that current point \*/  
           else  
           m1=10000;  
     }  
     if(m1!=10000||m!=10000) /\* if both wages of taxi is same then return the lowest index of taxi \*/  
      {  
            if(m<m1)  
                 {return (indx);}  
            else if(m>m1)  
                   {return (indx1);}  
            else  
           {  
           if(indx<indx1)  
                 {return (indx);}  
           else  
           {return (indx1);}  
           }  
     }  
   }  
 }  
return(777);/\*returning the error code if no taxi is availble\*/  
}  
  
int check2(char pic\_pnt,int p\_tme)   
/\*function to check the pick point\*/  
{  
int flag=0,min=10000,min\_ind,get; /\* set the value for min\*/  
for(int i=0;i<5;i++)  
{  
 if(pic\_pnt==t[i].cur\_pos)   /\*check the pickpoint of the customer with the current position of the taxi\*/  
 {  
    if(t[i].end\_time[(t[i].index-1)]<=p\_tme)   /\* check the previously allocated time of the customer with the current customer\*/  
    {  
     if(t[i].t\_wage<min) /\* calculate the minimium wage who earned that day \*/  
      {  
          min=t[i].t\_wage;  
          min\_ind=i;       /\*assigning minimum index\*/  
          flag=1;  
      }  
   }  
 }  
}  
if(flag==1)  
{return (min\_ind);}  
if(flag==0)            /\*returning minimum index to booking function\*/  
{return (-1);}  
return 0;  
}  
 void details()  
  {  
   int i,j;  
   pf("\t\t\t\tBOOKING DETAILS");  
   for(i=0;i<3;i++)  
   {  
     pf("\n\nTaxi %d details",i+1);   /\*print the details based on taxi\*/  
      pf("\nTotal wage:%d",t[i].t\_wage);  
     pf("\nc\_id  p\_pnt  d\_pnt  s\_tm  e\_tm   wage\n");  
     for(j=0;j<3;j++)/\*print the details based on cust id\*/  
     {  
       pf("%d      %c      %c    %d     %d     %d ",t[i].cust\_id[j],t[i].pic\_pnt[j],  
          t[i].drp\_pnt[j],t[i].str\_time[j],t[i].end\_time[j],t[i].wage[j]);  
       pf("\n");  
     }  
   }  
}