SECTION A: APPITUDE TEST 20 BITS
SECTION B: C BITS 20 BITS
SECTION C: 1 QUESTION (NOT A BIT)
IT IS LONG ANSWER
THE QUESTION IS:FIND THE NEXT PERMUTATION IN LEXICOGRAPHIC
ORDER FROM THE GIVEN PERMUTATION

NOTE: MAXIMUM QUESTIONS ARE FROM OLD QUESTION PAPERS

SECTION A:

1. THERE WERE 750 PEOPLE WHEN THE FIRST SONG WAS SUNG. AFTER EACH SONG 50 PEOPLE ARE LEAVING THE HALL. HOWMANY SONGS ARE SUNG TO MAKE THEM ZERO?
ANS:16

SECTION B:

1.
```
typedef struct{
   char *;
   nodeptr next;
   } * nodeptr;
```
what does nodeptr stand for?
ans:

2. 2 oranges,3 bananas and 4 apples cost Rs.15 . 3 ornages 2 bananas 1 apple costs Rs 10. what is the cost of 3 oranges, 3 bananas and 3 apples
ANs Rs 15.

3. `int *x[]();` means
Ans:expl: Elments of an array can't be functions.

4.
```
struct list{
int x;
struct list *next;
}*head;
```
the struct head.x =100
Ans: above is correct / wrong
expl: Before using the ptr type struct variable we have to give memory to that .
And also when ever the struct variable is ptr then we access the members by "->" operator.

5.o/p=?
```
int i;
i=1;
i=i+2*i++;
```

```
 printf(%d,i);
ans: 4

6. FILE *fp1,*fp2;
fp1=fopen("one","w")
fp2=fopen("one","w")
fputc('A',fp1)
fputc('B',fp2)
fclose(fp1)
fclose(fp2)}
a.error b. c. d.
ans: no error. But It will over writes on same file.

7.#include<malloc.h>
char *f()
{char *s=malloc(8);
strcpy(s,"goodbye")}
main()
{
char *f();
printf("%c",*f()='A');
o/p=?

8) #define MAN(x,y) (x)>(y)?(x):(y)
  {  int i=10;j=5;k=0;
  k= MAX(i++,++j)
  printf(%d %d %d %d,i,j,k)}
ans:10 5 0
9) a=10;b=5; c=3;d=3;
if(a<b)&&(c=d++)
printf(%d %d %d %d a,b,c,d)
else printf("%d %d %d %d a,b,c,d);
ans:

 : ..........................................
 10. what is o/p
  #include<stdarg.h>
  show(int t,va_list ptr1)
  {
  int a,x,i;
  a=va_arg(ptr1,int)
  printf("\n %d",a)
  }
  display(char)
 {int x;
  listptr;
  va_star(otr,s);
  n=va_arg(ptr,int);
  show(x,ptr);
  }
```

```
  main()
  {
  display("hello",4,12,13,14,44);
  }
  a) 13 b) 12 c) 44 d) 14

  ........................................
```

```
11.main()
{
printf("hello");
fork();
}
ans:

12.main()
{
int i = 10;
printf(" %d %d %d \n", ++i, i++, ++i);
}
ans:

13.#include<stdio.h>
main()
{
int *p, *c, i;
i = 5;
p = (int*) (malloc(sizeof(i)));
printf("\n%d",*p);
*p = 10;
printf("\n%d  %d",i,*p);
c = (int*) calloc(2);
printf("\n%d\n",*c);
}
ans:

14.#define MAX(x,y) (x) >(y)?(x):(y)
main()
{
        int i=10,j=5,k=0;
                k=  MAX(i++,++j);
              printf("%d..%d..%d",i,j,k);
                        }
ans:

15.#include <stdio.h>
main()
{
  enum _tag{ left=10, right, front=100, back};
  printf("left is %d, right is %d, front is %d, back is
```

```
%d",left,right,front,back);
}
ans:

16.main()
{
        int a=10,b=20;
        a>=5?b=100:b=200;
        printf("%d\n",b);
}
ans:


17.#define PRINT(int) printf("int = %d  ",int)
main()
{
int x,y,z;
x=03;y=02;z=01;
PRINT(x^x);
z<<=3;PRINT(x);
y>>=3;PRINT(y);
}
ans:

18.
#include<stdio.h>
main()
{
char s[] = "Bouquets and Brickbats";
printf("\n%c, ",*(&s[2]));
printf("%s, ",s+5);
printf("\n%s",s);
printf("\n%c",*(s+2));
}
ans:


19.
main()
 {
  struct s1
    {
    char *str;
    struct s1 *ptr;
     };
   static struct s1 arr[] = { {"Hyderabad",arr+1},
                              {"Bangalore",arr+2},
                              {"Delhi",arr}
                            };
               struct s1 *p[3];
```

```
                int i;
                for(i=0;i<=2;i++)
                        p[i] = arr[i].ptr;

                printf("%s\n",(*p)->str);
                printf("%s\n",(++*p)->str);
                printf("%s\n",((*p)++)->str);
        }
ans:


20.main()
{
char *p = "hello world!";
p[0] = 'H';


printf("%s",p);
}
ans:
```