Q1. What is the sizeof a pointer? Is it same across all platforms? How is it decided?

```
Q2.   main(){
                     char *p;
                     float *q;
                     double *r;
                 printf(“%lu %lu %lu\ n”,sizeof(p), sizeof(q),sizeof(r));
                 printf(“%lu %lu %lu\n”, sizeof(*p),sizeof(*q),sizeof(*r));
         }
```

```
Q3.    main( ) {
                      double  var;
                     char *cp=&var;//cosider the address of var is 5000
                     printf(“%lu %lu\n”,cp,cp+1);
}
```

```
Q4. main() {  char *cp=0; double *dp=0;
                printf(“%lu  %lu\n”,cp,dp);
                 ++cp;  ++dp;
             printf(“%lu %lu\n”,cp,dp);
             }
```

```
Q5. main() {   int  var=0x12345678; char *ptr=&var;
                 printf(“%x\n”,*ptr);
                  ptr++;
                 printf(“%x\n”,*(short int*)ptr);
         }
Q6. main()
    {
        char  ch=50;
        char  *ptr=&ch;
        printf(“%c\n”,--*ptr);
        printf(“%c\n”,ch);
    }
```

```
Q7.  void fun( int *p, int q)
    {
        printf(“in fun...%d  %d\n”,++*p,++q);
    }

    main() {
                 int x=10,y=20;
                 fun(&x,y);
                 printf(“in main...%d  %d\n”,x,y);
             }
```

```
Q8.
int *check(int, int);
int main()
{
int *c;
c = check(10, 20);
printf("%d", c);
return 0;
}
Int *check(int i,  int j)
{
int *p, *q;
p = &i;
q = &j;
if(i >= 45)
return (p);
else
return (q);
}
```

Q9.

```
main()
{
     unsigned int var=0x11223344;
     char *ptr=&var;
     printf(“%x\n”,*ptr++);
     printf(“%x\n”,++*ptr);
     printf(“%x\n”,*++ptr);
     printf(“%x\n”,--*ptr);
     printf(“%x\n”,*ptr--);
     printf(“%x\n”,var);
}
```

Q10.

```
     void  fun1(int  var)
     {
          printf(“%x\n”, var);
     }

     void fun2(int *ptr)
     {
          printf(“%x\n”, *ptr);
     }
main()
{
     float f=23.4;
      fun1(f);
      fun2(&f);
}
```

Q11. Write a program to verify the underlying CPU(where the program is to run) is little or big endian.

Q12. Write a program to print the binary equivalent of floating pointer data.
i)using one loop.(int* method)
ii) using two loops(char* method)