1.
```
#include<stdio.h>
int main(void)
{
    int a = 10;
    int *ptr = &a;

    printf(" %d %d ", a,++*ptr);
    printf(" %d %d ", a,*ptr++);

    return 0;
}
```

A. 11 11 11 11
B. 11 11 12 12
C. Compile time error
D. Run time error

Answer: A

2.
```
#include<stdio.h>
int main(void)
{
    int a[5] = {5,4,3,2,1};
    int *ptr = (int*)(&a+3-2);

    printf("%d %d", *(a+1), *(ptr-1));

    return 0;
}
```

A. 4 1
B. 2 1
C. 5 4
D. 1 1

Answer: A

3.

```
#include<stdio.h>
unsigned  char* convert(unsigned char *data)
{
    return (unsigned char*)data;
}
int main(void)
{
    unsigned char a = -64;
    unsigned char *data=NULL;

    data =convert(&a);
    printf("%d ",*data);

    return 0;
}
```

A. Complile time error
B. -64
C. 191
D. 192

Answer: D

4.

```
#include<stdio.h>
int main(void)
{
    int k = 5;
    int *p = &k;
    int **m = &p;

    **m = 6;
    ++--++*p;
    printf("%d\n", k);

    return 0;
}
```

A. compile time error
B. run time error
C. 7
D. 8

Answer: A

5.

```
#include <stdio.h>
void callbyAddress1(int *x)
{
    x=x+10;
}
void callbyAddress2(int *x)
{
    *x=*x+10;
}
int main(void)
{
    int a=10;
    printf(" %d ",a);
    callbyAddress1(&a);
    printf(" %d ",a);
    callbyAddress2(&a);
    printf(" %d ",a);
    callbyAddress1(&a);
    printf(" %d ",a);
    callbyAddress2(&a);
    printf(" %d ",a);

    return 0;
}
```

A. 10 10 20 20 30
B. 10 10 10 10 10
C. 10 20 30 40 50
D. compile time error

Answer: A

6.
```
#include<stdio.h>
int main( void )
{
    const int no=100;
    int  * const no_ptr = &no;
    printf(" %d",no);
    int value = *no_ptr ? ++*no : --*no   ;
    printf(" %d",value);
    return 0;
}
```
A. 100 100
B. 100 101
C. 100 garbage value
D. Compile time error
E. Run time error

Answer: D

7.
```
#include<stdio.h>
int main( void )
{
    unsigned char count = -2;
    void *counting = &count;
    ++*(signed char*)counting;
    printf("%hu ", *(char*)counting);
    ++*(signed char*)counting;
    printf("%hu\n", *(char*)counting);
    return 0;
}
```

A. 65535 0
B. -1 0
C. 4294967295 0
D. -429496729 0
E. run time error

Answer: A

8.
```
#include<stdio.h>
int main( void )
{
    void *ptr_name=NULL;
    char ch=115, *name="sunbeam";
    int j=117;

    ptr_name=&ch;
    printf("%c", *(char*)ptr_name);
    ptr_name=&j;
    printf("%c", *(int*)ptr_name);
    ptr_name=name;
    printf("%s",       (char*)ptr_name+2);

    return 0;
}
```

A. sunbeam
B. s117nbeam
C. 115unbeam
D. 115117unbeam

Answer: A

9.
```
#include<stdio.h>
void changeVal(short int *x)
{
    int i;

    for(i=0;i<sizeof(x);i++,x++)
    {
        *x+=2;
    }
    return ;
}
```

```
int main( void )
{
    short int  arr[]={1,2,3,4,5,6,7};
    changeVal(arr);
    printf("%d %d %d",arr[1],arr[3],arr[5]);
    return 0;
}
```

Note: conside 32 bit compiler setting

A. 4 6 6
B. 4 6 8
C. 4 4 4
D. Compiler error
E. 6 6 6

Answer: B

10.

```
#include<stdio.h>
int main( void )
{
    const int a = 4;
    int * const ptr = &a;*ptr = update(a);
    printf("a = %d ptr = %d ", a,--*ptr);
    printf("a = %d ptr = %d ", a,++*ptr);
    printf("a = %d ptr = %d ", a,--*ptr);
    return 0;
}
int update(int a)
{
    int value=(a+a*a+a);
    return value;
}
```

A. a = 23 ptr = 23 a = 24 ptr = 24 a = 23 ptr = 23
B. a = 63 ptr = 63 a = 64 ptr = 64 a = 63 ptr = 63
C. a = 24 ptr = 24 a = 25 ptr = 25 a = 24 ptr = 24
D. a = 64 ptr = 64 a = 65 ptr = 65 a = 64 ptr = 64
E. Compile time error

Answer: A

11.

```
#include<stdio.h>
int main( void )
{
    const int a = 10 , b = 20 ;
    const int * ptr = &a;
    int ** const ptr1 = &ptr;
    printf("\n a=%d *ptr=%d **ptr1=%d",a,*ptr,**ptr1);
       ptr = &b;
    **ptr1 = b;
    printf("\n a=%d *ptr=%d **ptr1=%d",a,*ptr,**ptr1);

    return 0;
}
```

A. a = 10 *ptr = 10 **ptr1=10
   a = 10 *ptr = 10 **ptr1=10

B. Run time error

C. Compile time error

D. a = 10 *ptr = 10 **ptr1=20
   a = 10 *ptr = 20 **ptr1=20

E. a = 10 *ptr = 10 **ptr1=10
   a = 10 *ptr = 20 **ptr1=20

Answer : E

12.

```
#include<stdio.h>
void changeVal2(int **x)
{
    **x*=2;
    return;
}
```

```
void changeVal1(int *x)
{
    *x/=5;
    return;
}
int main( void )
{
    int num1=100;
    int *ptr1=&num1;

    changeVal2(&ptr1);
    printf("\n num1=%d *ptr1=%d ",num1, *ptr1);

    changeVal1(ptr1);
    printf("\n num1=%d *ptr1=%d ",num1, *ptr1);

    changeVal2(&ptr1);
    printf("\n num1=%d *ptr1=%d ",num1, *ptr1);
    return 0;
}
```

A. num1=200 *ptr1=200
num1=40 *ptr1=40
num1=80 *ptr1=80

B. num1=20 *ptr1=20
num1=40 *ptr1=40
num1=8 *ptr1=8

C. num1=200 *ptr1=200
num1=400 *ptr1=400
num1=80 *ptr1=80

D. run time error

E. num1=200 *ptr1=200
num1=80 *ptr1=80
num1=40 *ptr1=40

Answer: A

13.
```
#include <stdio.h>
int x=100;
void callbyaddress(int **ptr_x)
{
    x=**ptr_x * **ptr_x / x;
}
int main( void )
{
    int x=10;
    int *ptr=&x;
    printf(" x = %d *ptr=%d ",x, *ptr);
    callbyaddress(&ptr);
    printf(" x = %d *ptr=%d ",x, *ptr);
    return 0;
}
```

A. x = 100 *ptr=100 x = 100 *ptr=100
B. x = 10 *ptr=10 x = 100 *ptr=100
C. x = 100 *ptr=100 x = 10 *ptr=10
D. x = 10 *ptr=10 x = 10 *ptr=10

Answer: D

14.
```
#include <stdio.h>
void modify(int ** const value)
{
    **value = 20;
    return ;
}
int main( void )
{
    const int value = 30;
    int * const ptr=&value;
    modify(&ptr);
    printf("value = %d\n", value);
    return 0;
}
```

A. value = 20
B. value = 30
C. compile time error
D. run time error

Answer: A

15.

```
#include<stdio.h>
int main( void )
{
    int num1 = 100, num2=150;
    int *p = &num1, **pp = &p;

    **pp = num2;
    ++**pp;
    printf("num1=%d *p=%d **pp=%d\n", num1, *p, **pp);
    --**pp;
    printf("num1=%d *p=%d **pp=%d\n", num1, *p, **pp);

    return 0;
}
```

A. num1=151 *p=151 **pp=151
   num1=150 *p=150 **pp=150

B. num1=150 *p=150 **pp=150
   num1=151 *p=151 **pp=151

C. run time error

D. comiple time error

E. num1=150 *p=150 **pp=150
   num1=150 *p=150 **pp=150

Answer: A

16.

```
#include<stdio.h>
#include<stdlib.h>
int compare1 (const void * a, const void * b)
{
  return ( *(int*)a - *(int*)b );
}
int compare2 (const void * a, const void * b)
{
  return ( *(int*)b - *(int*)a );
}
int main ( void )
{
  int arr[] = {99, 11, 33, 66, 99, 44};
  int n = sizeof(arr)/sizeof(arr[0]), i;

  qsort (arr, n, sizeof(int), compare2);
  for (i=0; i<n; i++)
      printf ("%d ", arr[i]);

  printf("\n");

  qsort (arr, n, sizeof(int), compare1);
  for (i=0; i<n; i++)
      printf ("%d ", arr[i]);

  return 0;
}
```

A. 99 66 44 33 22 11
   11 22 33 44 66 99

B. 11 22 33 44 66 99
   99 66 44 33 22 11

C. 11 22 33 44 66 99
   11 22 33 44 66 99

D. 99 66 44 33 22 11
   99 66 44 33 22 11

Answer : A

17.

```
#include<stdio.h>
int fun1(int n1, int n2)
{
    return n1+n2;
}
int fun2(int n2, int n1)
{
    return n1-n2;
}
void print(int n1, int n2,int (*fp)(int n1, int n2) )
{
    printf(" funptr = %d (*funptr) = %d" , fp, *fp);
    return;
}
int main( void )
{
    int val1=100, val2=200;
    int (*funptr)(int no1, int no2);

    funptr=fun2(val2,val2);
    print(val1, val2, funptr);
    funptr=fun1(val1, val1);
    print(10, 20, funptr);
    return 0;
}
```

A. run time error
B. funptr=300 (*funptr)=300 funptr= 100 (*funptr)= 100
C. funptr= 0 (*funptr)= 0 funptr= 200 (*funptr)= 200
D. funptr= 0 (*funptr)= 0 funptr= 20   (*funptr)= 20
E. compile time error

Answer: C