1.
```c
#include <stdio.h>
void print(double a[])
{
    int n=sizeof(a)/sizeof(*a)+sizeof(&a)-(a[7] /1.1f);
    int i;
    for (i = 0; i < n; i++)
        printf(" %.1lf ", a[i]);
    return;
}
int main( void )
{
    double arr[] = {1.1,2.2,3.3,4.4,5.5,6.6,7.7,8.8} ;
    print(arr);
    return 0;
}  //note :: consider 64 bit compilation.
```

A. 1.1
B. 1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8 0.0
C. 1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8 garbage
D. Compile time error

Answer: A

2.
```c
#include <stdio.h>
#define SIZE(arr) sizeof(arr) / sizeof(*arr);
void fun(int* arr, int n)
{
    *arr += *(arr + n - 1) += *arr;
}
void printArr(int* arr, int n)
{
    int i;
    for(i = 0; i < n; ++i)
        printf("%d ", arr[i]);
    return;
}
```

```
int main( void )
{
    int arr[] = {1, 2, 3};
    int size = SIZE(arr);

    fun(arr, size);
    printArr(arr, size);

    return 0;
}
// note consider 64 bit compilation.
```

A. 2 4 5
B. 5 2 4
C. 2 4 10
D. compile time error

Answer: B

3.

```
#include<stdio.h>
int main( void )
{
    int arr[3]={10,20,30,40};

    --2[arr];
    printf("%d",- --2[arr]);

    return 0;
}
```

A. -29
B. -28
C. Compiler error
D. 28

Answer: B

4.
```
#include <stdio.h>
int main(void)
{
    int arr[5]={5-3*0-1,10,15,20,25};

    printf("%8d",arr[arr[0]]);
    printf("%8d",arr[arr[1-1]]);
    printf("%8d",arr[arr[1*0]]);
    printf("%8d",arr[arr[0/1]]);

    return 0;
}
```

A. 25 25 25 25
B. 25 4 4 4
C. Compiler error
D. 4 4 4 4

Answer: A

5.
```
#include <stdio.h>
int main(void)
{
    int a[] = {45, 23, 43, 21, 98, 67};
    int *ptr = (int*)(&a+1);
    printf("%d ", *(ptr-sizeof('0')-1) );

    return 0;
}
```

A. 23
B. 21
C. 67
D. compile time error

Answer: A

6.

```
#include<stdio.h>
int main( void )
{
    int a[100]={-1},i=a[1];
    if(!(*(a+0+i)==i[0+i+a]))
    {
        printf("Welcome to Sunbeam @ Karad ");
    }
    else
    {
        printf("Welcome to Sunbeam @ Pune ");
    }
    return 0;
}
```

A. Complile time error
B. Welcome to Sunbeam @ Karad
C. Welcome to Sunbeam @ Pune
D. Run time error

Answer: C

7.

```
#include<stdio.h>
int main( void )
{
    int a[10]={1,2,3,4,1}, i,j,m;
    i=a[1]++ + ++a[2] + ++a[3];
    j=a[2]++ + --a[3] - a[4]--;
    m=++a[j];
    printf("%d,%d,%d",i,j,m);
    return 0;
}
```

A. 3,7,11
B. 1,7,11
C. 11,7,3
D. 11,7,1

Answer: D

8.

```
#include<stdio.h>
int main( void )
{
    double arr[]={1.2,2.3,3.4,4.5,5.6,6.7,7.8};

    float size=(float)(sizeof(arr)+1)/(sizeof(*arr));

    printf("%.4f",size);

    return 0;
}
//note :: consider 64 bit compilation.
```

A. 57.0000
B. 7.0000
C. 7.1250
D. 8.0000

Answer: C

9.

```
#include <stdio.h>
int main(void)
{
    char str[]="SunBeam IT Park";

    printf("%s\t%s\t%s",&str[8],&8[str],str+8);

    return 0;
}
```

A. IT Park IT Park IT Park
B. Compile time error
C. 32 32 IT Park
D. IT Park IT Park SunBeam IT Park

Answer: A

10.

```
#include <stdio.h>
#include<string.h>
int main(void)
{
    char* convert(char *s);

    char str1[] = "strings";
    char str2[] = "STRINGS";

    if (strcmp(convert(str1), convert(str2))==0)
        printf("%s Strings are equal %s", str1, str2);
    else
        printf("%s Strings are not equal %s", str1, str2);

    return 0;
}
char* convert(char *s)
{
    int i=0;
    while(*(s+i)!='\0')
    {
        if(*(s+i)>=65 && *(s+i)<=90)
            *(s+i)+= 32;
        else if (*(s+i)>=97 && *(s+i)<=122)
             i[s]-=32;

        i++;
    }

    return s;
}
```

A. STRINGS Strings are not equal strings
B. strings Strings are not equal STRINGS
C. Compile time error
D. STRINGS Strings are equal strings
E. strings Strings are equal strings STRINGS

Answer: A

11.
The correct statement to copy string literal constant “Hello” to string str is?

A. str=”Hello”
B. strcpy(“Hello”,str);
C. strcpy(str,”Hello”)
D. strcpy(str,’Hello’)

Answer: C

12.

```c
#include<stdio.h>
int main(void)
{
    char s[] = "Sunbeam", ch;
    int i=0;

    ch = s[i++];
    printf("%c", ch);

    ch = s[++i];
    printf("%c", ch);

    ch = ++i[s];
    printf("%c", ch);

    ch = i++[s];
    printf("%c", ch);

    return 0;
}
```

A. Snoo
B. Soon
C. Snoo
D. Sono

Answer: A

13.
```
#include <stdio.h>
int main(void)
{
    char dest[] = "Visual basic",src[] = "C++";
    puts(strcpy(&dest[7],src)-7);
    return 0;
}
```

A. Visual C++
B. c++
C. basic
C. Visual

Answer: A

14.
```
#include <stdio.h>
int main(void)
{
    char s[]="Sunbeam Pune";

    printf("%c ",*(&s[2]));
    printf("%s ",s+5);
    printf("%s ",s);
    printf("%c ",*(s+2));

    return 0;
}
```

A. n am Pune Sunbeam Pune n
B. nbeamPune amPune n
C. n a sunbeam Pune n
D. n a sunbeam pune n

Answer: A

15.

```
#include<stdio.h>
#include<string.h>
int main(void)
{
    char* str1 = "Sunbeam";
    char* str2= "Karad";
    char* str3=NULL;
    str3=strcat(str1, str2);
    printf("%s %s", str3, str1);
    return 0;
}
```

A. SunbeamKarad
B. It will terminates exit value -1
C. NULL Sunbeam
D. None of above

Answer: B

16.

```
#include<stdio.h>
#include<string.h>
int main( void )
{
    char str1[] = "Sunbeam", str2[] = "Sunbeam";
    if(!(str1==str2))
        printf("Equal");
    else
        printf("Unequal");
    return 0;
}
```

A. Equal
B. Unequal
C. Error
D. run time error
Answer : A

17.
Consider address of hellow as 4195764

```c
#include<stdio.h>
#include<string.h>
int main(void)
{
    printf("%u %s", &"Sunbeam", &"SunBeam");
    return 0;
}
```

A. 4195764 4195764
B. 4195764 SunBeam
C. 4195764 4195672
D. Sunbeam 4195764

Answer: B