1.

```
#include<stdio.h>
#define a 10
int main()
{
    #ifndef a
        #define a 15
    #else
        #undef a
        #define a 20
    #endif
    printf("a : %d",a);
   return 0;
}
```

A. a : 20
B. a : 10
C. a : 15
D.Compile time error
Answer: A

2.

```
#include<stdio.h>
#include<string.h>
int main()
{
  #define SUNBEAM "CDAC ATC \n"
  #define Sunbeam strlen(SUNBEAM)-5
  #ifdef SUNBEAM
    printf(SUNBEAM);
  #endif
  #ifdef SUNBEAM
    printf(Sunbeam+SUNBEAM);
  #endif

  return 0;
}
```

```
A.CDAC ATC
  ATC
B.CDAC ATC
  CDAC ATC
C.ATC
  ATC
D.CDAC ATC
  C ATC
```

Answer: A

3.

```
#include<stdio.h>
#define SQR(x)(x*x)
int main()
{
    int a, b=3;
    a = b*SQR(b+2);
    printf("%d\n", a);
    return 0;
}
```

a.25
b.11
c.33
d.75

Answer: C

4.

```
#include<stdio.h>
#define JOIN(s1,s2) printf("%s=%s %s=%s\n",#s1,s1,#s2,s2);
int main()
{
    char *str1="Sunbeam";
    char *str2="Pune";
    JOIN(str1, str2);
    return 0;
}
```

A.str1=Sunbeam str2=Pune
B.str1=Sunbeam Pune
C.str1=Pune str2=Sunbeam
D.Error: in macro substitution
Answer: A

5.

```
#include<stdio.h>
#define MIN(x, y) (x<y)? x : y;
int main()
{
    int x=3, y=4, z;
    z = MIN(x+y/2, y-1);
    if(z > 0)
        printf(" inside if %d\n", z);
    else
       printf("inside else %d\n", z);
    return 0;
}
```

A.inside if 4
B inside if 3
C.inside else 0
D.inside else -3
Answer: B

6.

```
#include<stdio.h>
#define str(x) #x
#define Xstr(x) str(x)
#define oper multiply
int main()
{
    char *opername = Xstr(oper);
    printf("print %s\n", opername);
    return 0;
}
```

A.Error: in macro substitution
B.Error: invalid reference 'x' in macro
C.print multiply
D.No output
Answer: C

7.

```
#include<stdio.h>
#define SUNBEAM SunBeam_Pune_Karad
int main()
{
    printf(" \"SUNBEAN\" \n");
    return 0;
}
```

A. "SUNBEAN"
B. SunBeam_Pune_Karad
C. SUNBEAM
D. “SunBeam_Pune_Karad”
Answer: A

8.

```
#include <stdio.h>
#define INCREMENT(x) ++x
int main()
{
    char *ptr = "SunbeamKarad";
    int x = 10;
    printf("%s", INCREMENT(ptr));
    printf("%3d",INCREMENT(x));
    return 0;
}
```

A.SunbeamKaradSunbeamKarad
B.unbeamKarad 11
C.ad 11
D.SunbeamKarad 11
Answer: b

9.

```
#include <stdio.h>
#define PRINT(i, limit) while (i < limit) \
                        { \
                           printf("\n Sunbeam @"); \
                           printf("Hinjawadi phase");\
                           printf(" %d ", ++i); \
                           i++; \
                        }
int main()
{
    int i = 1;
    PRINT(i, 3);
    return 0;
}
```

A.Sunbeam @Hinjawadi phase 2

B.Error: Cant write macros on multiple line.

C.Sunbeam @Hinjawadi phase 1
Sunbeam @Hinjawadi phase 2
Sunbeam @Hinjawadi phase 3

D.Sunbeam @Hinjawadi phase 1
Sunbeam @Hinjawadi phase 3

Answer:A

10. dont consider line numbers in codes.

```
1 #include <stdio.h>
2
3 int main (void)
4 {
5
6    printf ("This is line %d.\n", __LINE__-2);
7    return 0;
8 }
```

A.This is line 4.

b.This is line 6.

c.Compile Error
D.This is line followed by current line number.
Answer: A

11.

```
#include<stdio.h>
#include<stdlib.h>
int main ()
{
   char *title=NULL;
   title = (char *) malloc(15);
   strcpy(title, "C Programming");
   printf("String = %c", *title);
   free(title);
   title=0;
   return 0;
}
```

A.String = C Programming
B.Complile time error
C.String = C
D.Run time error
Answer:C

12.What will be the output of the following code on 64 bit compliation

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int *p=NULL;
    p = (int *)malloc(sizeof(int)*20);
    printf("%d bytes\n", sizeof(p));
    free(p);
    p=NULL;
    return 0;
}
```

A.4 bytes
B.2 bytes
C.8 bytes
D.80 bytes
Answer: C

13.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char *ptr=NULL;
    ptr = (char *)calloc(1,30);
    strcpy(ptr, "Sunbeam");
    printf("%d",ptr[9] );
    free(ptr);
    ptr=NULL;
    return 0;
}
```

A.nothing
B.0
C.Error: in free(ptr);
D.No error
Answer:b

14.
The .... function shrink or grow memory at runtime.
A.calloc( )
B.free( )
C.malloc( )
D.realloc( )
Answer: D

15.

```
#include<stdio.h>
#include<stdlib.h>
int main( void )
{
    int *ptr1=NULL, *ptr2=NULL;
    ptr1 = malloc(4);
    *ptr1 = 10;
    *ptr2 = free(ptr1);
    printf("%d\n",*ptr2);
    *ptr2=NULL;
    *ptr1=NULL;
    return 0;
}
```

A. 4
B. Run time error
C. Segmentation Fault
D. Compile time error
Answer: D