

1)

```
9  #include<stdio.h>
10 int main(){
11     int arr[]={6,12,18,24};
12     int x=0;
13     x=arr[1]+(arr[1]=2);
14     printf("%d",x);
15     return 0;
16 }
17
18
19
20
21
22
```

input

14

2)

```
9  #include<stdio.h>
10 #include<conio.h>
11 int f(int num){
12     return num*5;
13 }
14 void main(){
15     int i=3,val;
16     val=sizeof f(i)+ +f(i=1)+ +f(i-1);
17     printf("%d %d",val,i);
18 }
19
20
21
22
23
24
25
26
```

9 1

3)

```
9  #include<stdio.h>
10 #include<conio.h>
11
12 float avg(float x,float y,float z){
13     return (x+y+z)/3;
14 }
15 float avg(float,float,float);
16 void main(){
17     float p=1,q=2,r=-2,a;
18     a=avg(p,(q=4,r=-12,q),r);
19     printf("%f",a);
20 }
21
22
23
24
25
26
27
28
```

-2.333333


4)

```
9  #include <stdio.h>
10
11
12 int main() {
13     int x = 5;
14     int *p = &x;
15     printf("%d", ++(*p));
16     return 0;
17 }
18
19
20
21
22
23
24
25
26
27
28
```

6

5)

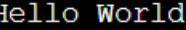
```
9  #include <stdio.h>
10
11 int main() {
12     int c[ ]={2.8,3.4,4,6.7,5};
13     int j,*p=c,*q=c;
14     for(j=0;j<5;j++) {
15         printf(" %d ",*c);
16         ++q; }
17     for(j=0;j<5;j++){
18         printf(" %d ",*p);
19         ++p; }
20
21 }
22
23
24
25
26
27
28
```



2 2 2 2 2 2 3 4 6 5

6)

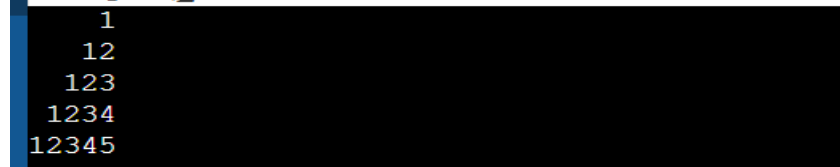
```
9  #include <stdio.h>
10 void display(char *string)
11 {
12     printf("%s",string);
13 }
14 int main() {
15     char string[]="Hello World";
16     display(string);
17
18 }
19
20
21
22
23
24
25
26
27
28
```



Hello World

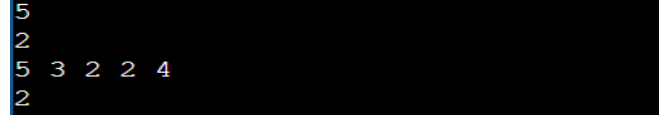
7)

```
1  #include <stdio.h>
2
3  int main() {
4      int n = 5;
5      int i, j, k;
6
7      for (i = 1; i <= n; i++) {
8          for (j = 1; j <= n - i; j++) {
9              printf(" ");
10         }
11         for (k = 1; k <= i; k++) {
12             printf("%d", k);
13         }
14         printf("\n");
15     }
16
17     return 0;
18 }
19
```




8)

```
1  #include <stdio.h>
2
3  int main() {
4      int N, X;
5      scanf("%d %d", &N, &X);
6      int arr[N];
7      for(int i=0; i<N; i++) {
8          scanf("%d", &arr[i]);
9      }
10     int index = -1;
11     for(int i=0; i<N; i++) {
12         if(arr[i] == X) {
13             index = i;
14             break;
15         }
16     }
17     printf("%d", index);
18     return 0;
19 }
20
```



9)


```
8
9  #include <stdio.h>
10 #include "string.h"
11 typedef struct stu1{
12     char name1[6];
13     char name2[6];
14     double marks;
15 }STU1;
16 void main(){
17     STU1 s1={"rohit","kumar",87.43},*p1;
18     char *p;
19     p1=&s1;
20     p=memchr(p1,'u',sizeof(STU1));
21     printf("%s",p);
22 }
23
24
25
26
27
28
```



The image shows a C program in a code editor. The code defines a struct 'stu1' with two character arrays 'name1' and 'name2' of size 6, and a double 'marks'. In the 'main' function, a struct variable 's1' is initialized with "rohit", "kumar", and 87.43. A pointer 'p1' is assigned the address of 's1'. Then, 'memchr' is used to find the first occurrence of the character 'u' in the memory starting from 'p1'. The 'printf' function prints the string starting from the found character, which is 'u', followed by 'mar', resulting in the output 'umar'.

10)

```
8
9  #include <stdio.h>
10 #define square(x) x*x
11 void main()
12 {
13     int i;
14     i = 64/square(4);
15     printf("%d",i);
16 }
17
18
19
20
21
22
23
24
25
26
27
28
```



The image shows a C program in a code editor. The code defines a macro 'square(x)' as 'x*x'. In the 'main' function, an integer 'i' is declared and assigned the value of 64 divided by 'square(4)'. Since 'square(4)' evaluates to 16, 'i' becomes 4. The 'printf' function prints the value of 'i', which is 4. However, the output shown in the image is 64, which suggests a misunderstanding of the macro's behavior or a typo in the code. Based on the code provided, the output should be 4.