Tut. Sheet 4

Solutions @Ayush Rathore

1. What is the output?

```
# include <stdio.h>
int main( )
{
int array[ 26 ], i ;
for ( i = 0 ; i <= 25 ; i++ )
{
array[ i ] = 'A' + i ;
printf ( "%d %c\n", array[ i ], array[ i ] ) ;
}
return 0 ;
}
```

65 A
66 B
67 C
68 D
69 E
70 F
71 G
72 H
73 I
74 J
75 K
76 L
77 M
78 N
79 O
80 P
81 Q

82 R
83 S
84 T
85 U
86 V
87 W
88 X
89 Y
90 Z

2.

```
# include <stdio.h>
int main( )
{
int size ;
scanf ( "%d", &size ) ;
int arr[ size ] ;
for ( i = 1 ; i <= size ; i++ )
{
scanf ( "%d", &arr[ i ] ) ;
printf ( "%d\n", arr[ i ] ) ;
}
return 0 ;
}
```

Error I undeclared

Else size input then array input/output

3

```
# include <stdio.h>
int main( )
{
int b[ ] = { 0, 20, 0, 40, 5 } ;
int i, *k ;
k = b ;
for ( i = 0 ; i <= 4 ; i++ )
{
printf ( "%d\n" *k ) ;
k++ ;
}
return 0 ;
}
```

error , *k

0
20
0
40
5

4

```
#include <stdio.h>
int
main ()
{
  int a[5] = { 5, 1, 15, 20, 25 };
  int i, j, k = 1, m;
  i = ++a[1];
  j = a[1]++;
  m = a[i++];
  printf ("%d %d %d\n", i, j, m);
}
```

3 2 15

5.

```
# include <stdio.h>
void jiaaysjo ( int , int )
int main( )
{
int p = 23, f = 24 ;
jiaaysjo ( &p, &f ) ;
printf ( "%d %d\n", p, f ) ;
return 0 ;
}
void jiaaysjo ( int q, int g )
{
q = q + q ;
g = g + g ;
}
```

function declaration ;

23 24

if want to correct

```
# include <stdio.h>
void jiaaysjo ( int *, int *)
int main( )
{
int p = 23, f = 24 ;
jiaaysjo ( &p, &f ) ;
printf ( "%d %d\n", *p, *f ) ;
return 0 ;
}
void jiaaysjo ( int *q, int *g )
{
*q = *q + *q ;
*g = *g + *g ;
}
```

46 48

6

```
# include <stdio.h>
int f ( int ) ;
int g ( int ) ;
int main( )
{
int x, y, s = 2 ;
s *= 3 ;
y = f ( s ) ;
x = g ( s ) ;
printf ( "%d %d %d\n", s, y, x ) ;
return 0 ;
}
int t = 8 ;
int f ( int a )

{
a += -5 ;
t -= 4 ;
return ( a + t ) ;
}
int g ( int a )
{
a = 1 ;
t += a ;
return ( a + t ) ;
}
```

6 5 6

7/

```
# include <stdio.h>
int g ( int ) ;
int main( )
{
int i, j ;
for ( i = 1 ; i < 5 ; i++ )
{
j = g ( i ) ;
printf ( "%d\n", j ) ;
}
return 0 ;
```

```
}
int g ( int x )
{
static int v = 1 ;
int b = 3 ;
v += x ;
return ( v + x + b ) ;
}
```

6
9
13
18

static remains after function ends. Hence value of v doesnt initialise everytime function is called.

8.

```
# include <stdio.h>
int main( )
{
int n[ 3 ][ 3 ] = {2, 4, 3,
6, 8, 5,
3, 5, 1
} ;

int i, *ptr ;
ptr = n ;
for ( i = 0 ; i <= 8 ; i++ )
printf ( "%d\n", *( ptr + i ) ) ;
return 0 ;
}
```

2
4
3
6
8

5
3
5
1