```
#include <stdio.h>
#include <stdlib.h>

#define ROWS 5
#define COLS 5

#define MAX 80
#define UPPER_ENCRYPT ((ch - 'A') + offset) % 26 + 'A'
#define LOWER_ENCRYPT ((ch - 'a') + offset) % 26 + 'a'

int main()
{
   printf("1. Write a program that creates an array containing the first 40 Fibonacci numbers.
\n");
   int n1=0,n2=1,n3,i;

   for(i=2;i<40;++i)//loop starts from 2 because 0 and 1 are already printed
      {
         n3=n1+n2;
         printf(" %d",n3);
         n1=n2;
         n2=n3;
      }
```

```
1. Write a program that creates an array containing the first 40 Fibonacci numbers.
 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887 9227465 14930352 24157817 39088169 63245986
2. Write a program that declares a two dimensional array that will store the hourly temperature for 30 days.
Day 1:
00:00  01:00  02:00  03:00  04:00  05:00
06:00  07:00  08:00  09:00  10:00  11:00
12:00  13:00  14:00  15:00  16:00  17:00
18:00  19:00  20:00  21:00  22:00  23:00

Day 2:
00:00  01:00  02:00  03:00  04:00  05:00
06:00  07:00  08:00  09:00  10:00  11:00
12:00  13:00  14:00  15:00  16:00  17:00
18:00  19:00  20:00  21:00  22:00  23:00

Day 3:
00:00  01:00  02:00  03:00  04:00  05:00
06:00  07:00  08:00  09:00  10:00  11:00
12:00  13:00  14:00  15:00  16:00  17:00
18:00  19:00  20:00  21:00  22:00  23:00

Day 4:
00:00  01:00  02:00  03:00  04:00  05:00
06:00  07:00  08:00  09:00  10:00  11:00
12:00  13:00  14:00  15:00  16:00  17:00
18:00  19:00  20:00  21:00  22:00  23:00

Day 5:
00:00  01:00  02:00  03:00  04:00  05:00
06:00  07:00  08:00  09:00  10:00  11:00
12:00  13:00  14:00  15:00  16:00  17:00
18:00  19:00  20:00  21:00  22:00  23:00

Day 6:
00:00  01:00  02:00  03:00  04:00  05:00
06:00  07:00  08:00  09:00  10:00  11:00
12:00  13:00  14:00  15:00  16:00  17:00
18:00  19:00  20:00  21:00  22:00  23:00

Day 7:
00:00  01:00  02:00  03:00  04:00  05:00
06:00  07:00  08:00  09:00  10:00  11:00
12:00  13:00  14:00  15:00  16:00  17:00
18:00  19:00  20:00  21:00  22:00  23:00
```

```
   printf("\n2. Write a program that declares a two dimensional array that will store the hourly
temperature for 30 days. \n");

   int j;
```

```
        for (i = 0; i < 30; i++) {
                printf("Day %d:\n", i + 1);
                for (j = 0; j < 24; j++) {
                        printf("%.2d:00  ", j);
                        if ((j + 1) % 6 == 0) {
                                printf("\n");
                        }
                }
                printf("\n");
        }
```

printf("\n3. Use the array in question 2 and write a program to compute the average temperature. \n");

```
        int n[35][30];
   int av;
   for(int i=1; i<=30; i++)
      {
         for(int j=0;j<24;j++)
         {
            printf("temp on day %d at %02d:00 : ", i, j);
            scanf("%d",&n[i][j]); av+=n[i][j];
         }
      }
```

```
Temp on day 30 at 17:00: 3
Temp on day 30 at 18:00: 5
Temp on day 30 at 19:00: 5
Temp on day 30 at 20:00: 5
Temp on day 30 at 21:00: 5
Temp on day 30 at 22:00: 4
Temp on day 30 at 23:00: 4
The average temperature: 34
```

```
   printf("Average temperature is: %d", av/(30*24));
```

printf("\n4. Write a program that reads a 5 x 5 array of integers and prints the row and column sums. \n");

```
        int a[ROWS][COLS], row_totals[ROWS] = {0}, col_totals[COLS] = {0};

        for (int i = 0; i < ROWS; i++) {
                printf("Enter row %d: ", i + 1);
                scanf("%d%d%d%d%d", &a[i][0], &a[i][1], &a[i][2], &a[i][3], &a[i][4]);
```

```
}


for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
                row_totals[i] += a[i][j];
                col_totals[j] += a[i][j];
        }
}

printf("\nRow totals:");
for (int i = 0; i < ROWS; i++) {
        printf(" %d", row_totals[i]);
}

printf("\nColumn totals:");
for (int i = 0; i < COLS; i++) {
        printf(" %d", col_totals[i]);
}
printf("\n");
```

```
4. Write a program that reads a 5 x 5 array of integers and prints the row and column sums.
Enter row 1: 11 12 13 14 15
Enter row 2: 25 24 23 22 21
Enter row 3: 31 32 33 34 35
Enter row 4: 45 44 43 42 41
Enter row 5: 25 50 75 100 125

Row totals: 65 115 165 215 375
Column totals: 137 162 187 212 237
```

```
printf("\n5. Write a program that encrypts a message by replacing each letter with the letter\n");
printf("that is n letters ahead of it in the alphabet (wrapping round to A if you go past Z) e.g. if
n=3 then A->D, B->E, C->F etc. \n");
        char s[100];
        scanf("%s", &s);
                for(int i=0;i<strlen(s);i++)
                {
                        if (s[i]=='X')
                        {
                                s[i]=='A';
                        }
                        else if (s[i]=='Y')
                        {
                                s[i]=='B';
                        }
                        else if (s[i]=='Z')
```

```
                {
                        s[i]=='C';
                }
                else if (s[i]=='x')
                {
                        s[i]=='a';
                }
                else if (s[i]=='y')
                {
                        s[i]=='b';
                }
                        else if (s[i]=='z')
                {
                        s[i]=='c';
                }
                else
                {
                        s[i]=s[i]+3;
                }
        }
        printf("%s", s);
    return 0;
}
```

```
5. Write a program that encrypts a message by replacing each letter with the letter that
is n letters ahead of it in the alphabet (wrapping round to A if you go past Z) e.g. if
n=3 then A->D, B->E, C->F etc.

Soekarno
Vrhnduqr

Process returned 0 (0x0)   execution time : 0.026 s
Press any key to continue.
```