Annex-I

Use these strips to play Scytale

Annex-II

Encrypted text to work with at the end of Module-II
Hint!

- It is a permutation cipher
- ‘a’ has been permuted to ‘z’, so you need to map z → a to get the plain text back

gsv hgifxgfiv zmw lkvizgrlm lu gsv vbv rh

evib hrnrozi gl zm vovxgilmrx xznviz,zmw rg

rh mzgfizo gl wrhxfhh gsvn gltvgsvi. ylgs

ziv yzhvw lm gdl nzqli xlnklmvmgh: z ovmh

zhhvnyob, zmw zm rnztrmt hvmhli. gsv ovmh

zhhvnyob xzkgfivh z kligrlm lu gsv ortsg

vnzmzgrmt uiln zm lyqvxg, zmw ulxfh rg

lmgl gsv rnztrmt hvmhli. gsv rnztrmt hvmhli

gsvm gizmhulinh gsv kzggvim lu ortsg rmgl z

rwvl hrtmzo, vrgsvi vovxgilmrx li mvfizo.

In Cipher	to	In Plain
a	to	
b	to	
c	to	
d	to	
e	to	
f	to	
g	to	
h	to	
i	to	
j	to	
k	to	
l	to	
m	to	

In Cipher	to	In Plain
n	to	
o	to	
p	to	
q	to	
r	to	
s	to	
t	to	
u	to	
v	to	
w	to	
x	to	
y	to	
z	to	a

Annex-III

Part of the EASCII table

Binary	Value
00100000	SP (Space)
00100001	!
00100010	"
00100011	#
00100100	$
00100101	%
00100110	&
00100111	'
00101000	(
00101001	)
00101010	*
00101011	+
00101100	,
00101101	-
00101110	.
00101111	/
00110000	0
00110001	1
00110010	2
00110011	3
00110100	4
00110101	5
00110110	6
00110111	7
00111000	8
00111001	9
00111010	:
00111011	;
00111100	<
00111101	=
00111110	>
00111111	?
01000000	@
01000001	A
01000010	B
01000011	C
01000100	D
01000101	E
01000110	F
01000111	G
01001000	H
01001001	I
01001010	J
01001011	K
01001100	L
01001101	M
01001110	N
01001111	O

Binary	Value
01010000	P
01010001	Q
01010010	R
01010011	S
01010100	T
01010101	U
01010110	V
01010111	W
01011000	X
01011001	Y
01011010	Z
01011011	[
01011100	\
01011101	]
01011110	^
01011111	_
01100000	`
01100001	a
01100010	b
01100011	c
01100100	d
01100101	e
01100110	f
01100111	g
01101000	h
01101001	i
01101010	j
01101011	k
01101100	l
01101101	m
01101110	n
01101111	o
01110000	p
01110001	q
01110010	r
01110011	s
01110100	t
01110101	u
01110110	v
01110111	w
01111000	x
01111001	y
01111010	z
01111011	{
01111100	|
01111101	}
01111110	~
01111111	DEL (delete)

Work Sheet for Module-4

C(Cipher)	F								{								y							
C (Cipher in bin.)	0	1	0	0	0	1	1	0	0	1	1	1	1	0	1	1	0	1	1	1	1	0	0	1
? (What?)																								
P																								
P																								

5								g								s								`							
0	0	1	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	1	0	0	0	0	0

>								}								a								>							
0	0	1	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	1	0	0	0	0	1	0	0	1	1	1	1	1	0

.								5								q								;							
0	0	1	0	1	1	1	0	0	0	1	1	0	1	0	1	0	1	1	1	0	0	0	1	0	0	1	1	1	0	1	1

5								}								q								z							
0	0	1	1	0	1	0	1	0	1	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	1	1	1	1	0	1	0

}								w								;															
0	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	0	0	1	1	1	0	1	1								

Key

0	0	0	1	0	1	0	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0

Annex-V

Work Sheet for Module-6 (we are working in **<u>mod (29)</u>**)

$\mathbf{2^x}$	**Working**	**Result**
2^1		2
2^2		4
2^3		8
2^4		16
2^5	32=3	3
2^6	$(2^5)2 = 3x2$	6
2^7		
2^8		
2^9		
2^{10}		
2^{11}		
2^{12}		
2^{13}		
2^{14}		
2^{15}		
2^{16}		
2^{17}		
2^{18}		
2^{19}		
2^{20}	$(2^5)^4 = (3)^4 = (9)^2 = 81$	23
2^{21}		
2^{22}		
2^{23}		
2^{24}	$(2^4)(2^{20}) = 16x23 =$ 8x2x23=8x17=4x2x17=4x5	20
2^{25}		
2^{26}		
2^{27}		
2^{28}		

$\mathbf{2^x}$	**Working**	**Result**
7^1		
7^2		
7^3		
7^4		
7^5		
7^6		
7^7		
7^8		
7^9		
7^{10}		
7^{11}		
7^{12}		
7^{13}		
7^{14}		
7^{15}		
7^{16}		
7^{17}		
7^{18}		
7^{19}		
7^{20}		
7^{21}		
7^{22}		
7^{23}		
7^{24}		
7^{25}		
7^{26}		
7^{27}		
7^{28}		

Annex-VI

//C++ code to get the power of a number mod (p)

```
#include <iostream>
#include <conio.h>
#define N 100
using namespace std;

void main()
{
      long int p;
      long int g;
      long int a;
      long int b;

cout << "Enter the value of prime:";
cin >> p;

cout <<endl;
cout <<"Enter the number whose power you want to raise:";
cin >> g;

cout <<endl<<"Enter the power you want to raise:";
cin >> a;
cout <<endl;

      b=g;
      for(int i = 1; i < a; i++)
          b=(b*g)%p; //Comments:Current value of b is divided by p and
                     //the remainder is assigned to it
      cout << g << " Raised to the power " << a <<" is " << b<< endl;
}
```