Question 1

Source code:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
	int i, output;
	int A[8] = { 0, 0, 0, 0,1,1,1,1 };
	int B[8] = { 0, 0, 1, 1,0,0,1,1 };
	int C[8] = { 0, 1, 0, 1,0,1,0,1 };

	for (i = 0; i < 8; i++) {
	// using ‘&&’ Operator

	output = A[i] * B[i] * C[i];
	printf("\n %d AND %d AND %d = %d",A[i], B[i],C[i], output);

	}
}
```

Output:

```
 0 AND 0 AND 0 = 0
 0 AND 0 AND 1 = 0
 0 AND 1 AND 0 = 0
 0 AND 1 AND 1 = 0
 1 AND 0 AND 0 = 0
 1 AND 0 AND 1 = 0
 1 AND 1 AND 0 = 0
 1 AND 1 AND 1 = 1
```

Question 2

Source code:

```
#include <stdio.h>

int roman(int year){

		if (year>=1000)
		{
			printf("m");
			roman(year-1000);
		}
		else if(year>=500)
		{
			printf("d");
			roman(year-500);
		}

		else if(year>=100)
		{
			printf("c");
			roman(year-100);
		}
		else if(year>=50){
			printf("l");
			roman(year-50);
		}
		else if(year>=10){
			printf("x");
			roman(year-10);
		}
		else if(year>=5){
			printf("v");
			roman(year-5);
		}
		else if(year>=1){
			printf("i");
			roman(year-1);
		}
	}
int main()
{
	int year;
	printf("Enter the year you want to convert: \n");
	scanf("%d",&year);

	roman(year);
}
```

Output:

```
Enter the year you want to convert:
1998
mdcccclxxxxviii
```

Question 3

Source code:

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

int main()
{
	int *ptr;	//declaration of integer pointer
	int n;	//to store array limit
	int i;		//loop counter
	int sum;	//to store sum of all elements

	printf("Enter limit of the array: ");
	scanf("%d",&n);

	//declare memory dynamically
	ptr=(int*)malloc(n*sizeof(int));

	//read array elements
	for(i=0;i<n;i++)
	{
		printf("Enter element %d: ",i+1);
		scanf("%d",(ptr+i));
	}

	//print array elements
	printf("\nEntered  elements are:\n");
	for(i=0;i<n;i++)
	{
		printf("%d\n",*(ptr+i));
	}

	//calculate sum of all elements
	sum=0;		//assign 0 to replace garbage value
	for(i=0;i<n;i++)
	{
		sum+=*(ptr+i);
	}
	printf("Sum of array elements is: %d\n",sum);

	//free memory
	free(ptr);	//hey, don't forget to free dynamically allocated memory.

	return 0;
}
```

Output:

```
Enter limit of the array: 4
Enter element 1: 3
Enter element 2: 12
Enter element 3: 7
Enter element 4: 8

Entered  elements are:
3
12
7
8
Sum of array elements is: 30
```