State machine:

know as finite-state Machine a finite automation.

Computation modle that represents a system with finite number of state.

Key concepts:

- States: represent different conditions orode that system can be in.
- Transition: define how the system move from one state to another triggered by Event or input.
- initial states-the state the system state when it first activated .
  Finite:the number State is limited.

State machine (modle behaviour)

State: different modle or condition.
Events: things that cause change.
Transition:move between stage.
Action:thing done during Transition or in state.

Events:

An event is an action or occurance that a program can respond to.

C doesn't have built in event handling,but we can can simulate events using FUNCTION POINTER.

Events Example:

Events handler without parameters:

```
#include<stdio.h>
void greet(){
printf("hello");
}
typedef void(*eventhandler)();
int main(){
eventhandler event;
event=greet;
event();
}
```

Events handler with parameters:

```
#include<stdio.h>
typedef void(*eventhander)(int);
void on_event(int value){
printf("value:%d\n",value);
}
```

```
int main(){
eventhander handler1=on_event;
handler1(10);
}
```

Events (array handling)

```
#include <stdio.h>
typedef void (*eventhandler)(int);
void handler_a(int value) {
printf("[A] Value: %d\n", value);
}

void handler_b(int value) {
printf("[B] Value: %d\n", value);
}

int main() {
eventhandler handlers[] = {handler_a, handler_b};
int num_handlers = sizeof(handlers) / sizeof(handlers[0]);
for (int i = 0; i < num_handlers; i++) {
handlers[i](i + 1);
}

return 0;

}
```

Pointer:

A pointer is a variable that stores the memory address of another variable.

pointer is declared by specifying its name and type, just like simple variable declaration but with an asterisk (*) symbol added before the pointer's name.

Syntax:data_type* name

Example :

```
#include<stdio.h>
int main(){
int num = 100;
int *ptr = #
printf("number:%d\n",num);
printf("number:%p\n",&num);
printf("number%p\n",ptr);
printf("number:%d\n",*ptr);
}
```

Array:

An array in C is a fixed-size collection of similar data items stored in contiguous memory locations.

It can be used to store the collection of primitive data types such as int, char, float, etc., as well as derived and user-defined data types such as pointers, structures, etc.

Example:

```
#include<stdio.h>
int main(){
int num[5]={1,2,3,4,5};
printf("array elements:\n");
for(int i=0;i<5;i++){
printf("number[%d]=%d\n",i,num[i]);
}
}
```