* stack data pointer → top

Tutorial-: 01)-:

Q-: 01) What is circular queue?

Data structure that efficiently manages a fixed-size collection of elements

a circular queue also known as ring buffer. It a data structure that follows FIFO principle. It is implemented as an array queue with the fixed size, where the last position is connected to the first position forming a circular behavior.

Q-: 02) What is the characteristics of circular queue?

- circular structure.
- fixed size.
- front and rear pointers.
- enqueue operation.
- dequeue operation.

pointers → (front and rear)
- the circular behavior
- fixed size queues
- the circular queues are initialized with the fixed size which prevent them from drawing dynamically

i) In circular queue 2 pointers (front and rear) maintains the order of the circular queue.

ii) When the data structure has reach its capacity the next element will be inserted into a vacant position in the front of data structure

Q-: 03). Give applications of circular queue.

- memory management → circular queue is used in memory management.
- process scheduling (operating/CPU scheduling) → A CPU uses a queue to shedule processes.
- taffic system (management) → queues are also used in taffic system.

Q-: 04) What is algorithm of circular queue?

initialize the queue.

enqueue. (insertion)

dequeue (delection)
get the front element.
check if the queue is empty.
check if the queue is full.

Q-8 05) Compare and contrast linear queue and circular queue.

linear queue	circular queue.
* less efficient	more efficient.
* requires more memory space.	requires less memory space.
* it arrange data in linear order.	it arrange data in circular order.
* we can easily fetch out the peek value	we cannot fetch out the peek value easily.

Q-: 00) Write a simple program of circular.

```
int main()
{
  double radius, circumference, area;

  printf("enter the radius of the circle");
  scanf("%lf", &radius);

  circumference = 2 * PI * radius;
  area = PI * redius * radius

  printf("circumference: %.2lf\n", circumfere
                                    -nce);

  printf("area: %.2lf\n", area);

  return 0;
}
```