**1.**

**ابتدا ورودی را تبدیل به یک آرایه ی یک بعدی میکنیم که اندیس های زوج دیتا هستند و اندیس های فرد عدد لینک را مشخص می کنند . سپس با struct ای که تعریف کردم که دارای data و dlink و next است خانه های زوج این آرایه را در data ی این struct می گذارم و خانه های فرد این آرایه را در dlink این struct می گذارم و قسمت next را به خانه ی بعدی اشاره میدهم این کار تا زمانی انجام می شود که مقدار اندیس خانه ی بعدی از اندیس خانه ی قبلی بزرگتر است**

struct Node

{

int data;

int dlink;

Node \*next;

};

**توابع:**

**Input: برای تبدیل ورودی به یک ارایه ی یک بعدی**

**Addtolist : برای ذخیره کردن آرایه در لینک لیست وحذف لوپ**

**فقط توجه داشته باشید فرمت ورودی آخرش ; دارد**

**مرتبه زمانی o(n) است**

**2.**

**Kadane’s algorithm :**

**برای آرایه ی یک بعدی میتوان از آن استفاده کرد که ایده ی آن این است که در هر ردیف با توجه به right و left ها ماکسیمم جمع را پیدا میکند که بعد از آن با top و bottom ردیف هایی که باهم بیشترین جمع را خواهند داشت پیدا می کند و نمایش می دهد. برای اینکار ماکسیمم جمع هر ردیف را در temp[i] میریزیم . برای بدست آوردن ماکسیمم جمع هر دفعه جمع بدست آمده را با ماکسیمم جمعی که داریم مقایسه می کنیم و در صورت نیاز آن را تغییر می دهیم.**

**توابع :**

**Input :برای تبدیل ورودی به یک ارایه ی دوبعدی**

**Kardane: الگوریتم محاسبه ی maxsum برای آرایه ی یک بعدی**

**maxSum : برای محاسبه ی بزرگترین submatrix**

**فقط توجه داشته باشید فرمت ورودی آخرش ; دارد**

**مرتبه زمانی : o(n^3)**