

تعتبر المتجهات (Arrays) و القوائم المترابطة (Linked Lists) ، و بنى المعطيات الأخرى مثل المكس و الرتل بنى ملائمة لتنظيم البيانات التي نجدتها في تطبيق قاعدة البيانات . و هي عادة ما تستخدم في السجلات الوظيفية و سجلات الجرد و البيانات المالية ، كما أن استخدام هذه البنى من شأنه أن يساهم في تيسير عملية الوصول إلى البيانات و التعامل معها سواء بالإدخال أو بالحذف أو البحث عن عناصر معينة .

مفهوم - فكرة ال Stack هي عبارة عن خط انتظار لمجموعة من البيانات، ما يميز هذا الخط بأنه مفتوح من اتجاه واحد فقط، أي أن البيانات تدخل وتخرج من بوابة واحدة.

يطلق على stack مصطلح LIFO (last in first out) وهذا الرمز يعني بأن آخر عنصر دخل إلى stack هو أول عنصر يغادرها، وهذا بديهي بما أنه لا يوجد لدينا الى فتحة واحدة لإدخال وإخراج العناصر .

يمكن تشبيه المكس بحاملة الأقراص المدمجة فأخر قرص تقوم بوضعه هو أول قرص تقوم بإخراجه، أو كمجموعة كتب متراسة فوق بعضها البعض .

مفهوم queue - أو الرتل هو بنية معطيات مجردة مكونة من مجموعة تحتوي على عدد من العناصر التي يتم الحفاظ على ترتيبها وفق نظام FIFO ومعناها First In First Out.

ال Queue أو نظام الرتل له نفس فكرة عمل ال Stack ولكن هناك مكانين يخرج منهم البيانات وتقوم بإضافة البيانات بواسطة () enqueue وإزالة البيانات بواسطة () dequeue والترتيب الخاص به يعمل بنظام FIFO كما أوضحنا وهي تعنى العنصر الذي يدخل في البداية هو الذي يخرج أولاً. First In First Out.

أقرب مثال على الرتل هو قوائم الانتظار أو ال Waiting List حيث نجد أن أول من يدخل هو أول من يخرج و آخر من يدخل هو آخر من يخرج وهذا النوع من الترتيب يُعرف اختصاراً بـ FIFO أي First In First Out.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node
5  {
6  int data;
7  node *next;
8  };
9  typedef struct node node_t; 10
11 node_t *head = NULL; 12
```

تعريف العقدة:

```
13 void push(int n)
14 {
15 node_t *newNode = (node_t *)malloc(sizeof(node_t));
16 newNode->data = n;
17 newNode->next = NULL; 18
19 if(head == NULL) {
20 head = newNode;
21 return;
22 }
23
24 node_t *cur = head;
25 while(cur) {
26 if(cur->next==NULL) {
27 cur->next = newNode;
28 return;
29 }
30 cur = cur->next; 31 }
32 }
33
```

تعريف تابع اضافة عنصر الى الرتل:

تابع النزع من الرتل:

```
34 void pop()  
35 {  
36 if(head==NULL) return;  
37 node_t *tmp = head;  
38 head = head->next;  
39 free(tmp);  
40 }  
41
```

تابع عرض الرتل:

```
42 void display()  
43 {  
44 node_t *cur = head;  
45 while(cur) {  
46 printf("%3d",cur->data);  
47 cur = cur->next;  
48 }  
49 printf("\n");  
50 }  
51
```

التابع الرئيسي:

```
52 int main()  
53 {  
54 push(1);push(2);push(3);push(4);push(5);display();  
55 pop();display();  
56 pop();display();  
57 pop();display();  
58 pop();display();  
59 pop();display();  
60 return 0;  
61 }  
62
```

```
1  #include <iostream>
2  using namespace std;
3
```

صف الرتل:

```
4  class Queue
5  {
6  public:
7  Queue();
8  ~Queue();
9  void push(int);
10 int pop();
11 void print();
12 private:
```

تعريف العقدة:

```
13 typedef struct Node {
14     Node *next;
15     int data;
16 } NODE;
17 NODE* head;
18 };
19
```

التابع الباني:

```
20 Queue::Queue()
21 {
22     head = NULL;
23 }
24
```

التابع الهادم:

```
25 Queue::~~Queue()
26 {
```

```
27  if(head == NULL) return;
28  NODE *cur = head;
29  while(cur) {
30  Node *ptr = cur;
31  cur = cur->next;
32  delete ptr;
33  }
34  }
35
```

تابع اضافة عنصر الى الرتل:

```
36  void Queue::push(int n)
37  {
38  if(head == NULL) {
39  head = new NODE;
40  head->data = n;
41  head->next = NULL;
42  return;
43  }
44  NODE *cur = head;
45  while(cur) {
46  if(cur->next == NULL) {
47  NODE *ptr = new NODE;
48  ptr->data = n;
49  ptr->next = NULL;
50  cur->next = ptr;
51  return;
52  }
53  cur = cur->next; 54 }
55  }
56
```

تابع طباعة رتل:

```
57  void Queue::print()
58  {
```

```
59  if(head==NULL) return;
60  Node *cur = head;
61  while(cur) {
62  cout << cur->data << " ";
63  cur = cur->next;
64  }
65  cout << endl;
66  }
67
```

تابع نزع عنصر من الرتل:

```
68  int Queue::pop()
69  {
70  if(head == NULL) {
71  cout << "empty estack!" << endl;
72  return NULL;
73  }
74  NODE *tmp = head;
75  int value = head->data;
76  if(head->next) {
77  head = head->next;
78  }
79  // pop the last element (head)
80  else {
81  delete tmp;
82  head = NULL;
83  }
84  cout << "pop: " << value << endl;;
85  return value;
86  }
87
```

التابع الرئيسي:

```
88  int main()
```



```
89  {
90  Queue *que = new Queue();
91  que->push(10);
92  que->push(20);
93  que->push(30);
94  que->push(40);
95  que->push(50);
96  que->print();
97  que->pop();que->print();
98  que->pop();que->print();
99  que->pop();que->print();
100 que->pop();que->print();
101 que->pop();que->print();
102 que->pop();que->print();
103 return 0;
104 }
105
```