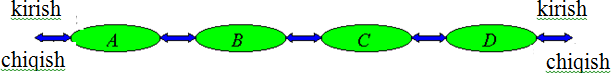
**Deklar**

Dek so’zi (*DEQ* - Double Ended Queue) ingliz tilidan olingan bo’lib 2 ta chetga ega navbat degan ma‟noni bildiradi. Dekning o’ziga xos xususiyati shuki, unga elementlar har ikkala tomondan – chapdan va o’ng tomondan kiritilishi va chiqarilishi mumkin (2.3-rasm).



2.3-rasm. Dek tuzilmasi

***Dek ustida bajariladigan amallar:***

1. Chapdan element kiritish.
2. O’ngdan element kiritish.
3. Chapdan element chiqarish.
4. O’ngdan element chiqarish.
5. Dek bo’shligini tekshirish.
6. Dek to’laligini tekshirish.

***C++ tilida dekni statik ko’rinishda, ya’ni bir o’lchashhamli massiv ko’rinishida amalga oshirishga misol*:** Berilayotgan butun sonlar ketma-ketligining 1-yarmini dekning chap tomonidan, qolgan yarmini dekning o’ng tomonidan kiriting. Dekning elementlarini bir safar chapdan, bir safar o’ngdan juftlikka tekshirib, toq elementlari o„chirilsin.

**Algoritm**

1. Dekka nechta element kiritilishi aniqlanadi – n, i=0.
2. i++; agar i<n bo’lsa va dek to„lmagan bo’lsa, 3-qadamga, aks holda 4-qadamga o„tiladi.
3. Agar i<n/2 bo’lsa, navbatdagi element chapdan, aks holda, ya’ni i>n/2 bo’lsa, dekning o’ng tomonidan kiritiladi, 2-qadamga o’tish.
4. Agar dek bo’sh bo„lmasa, chapdan element chiqarib olamiz. Agar element juft bo’lsa, b[] massivga joylaymiz. 5-qadamga o„tiladi. Agar dek bo’sh bo’lsa, 6- qadamga o’tish.
5. Agar dek bo’sh bo„lmasa, o’ngdan element chiqarib olamiz. Agar element juft bo’lsa, b[] massivga joylaymiz. 5-qadamga o„tiladi. Agar dek bo’sh bo’lsa, 6- qadamga o’tish.
6. b[] massiv elementlarini dekka o’ng tomondan kiritamiz.
7. Dek tarkibini ekranga chiqaramiz.

**Dastur kodi** *#include <cstdlib> #include <iostream> using namespace std; int a[10],n,R=0; bool isEmpty(){*

*if(R==0) return true; else return false;*

*bool isFull(){*

*if(R>=10) return true; else return false;*

*}*

*int kirit\_left(int s){*

*if(isFull()){cout<<"\ndek to'ldi";n=R;return EXIT\_SUCCESS;} for(int i=R;i>0;i--)*

*a[i]=a[i-1]; a[0]=s;R++;*

*}*

*int olish\_left(){*

*if(isEmpty()){cout<<"\ndek bo'sh";return EXIT\_SUCCESS;} int t=a[0];*

*for(int i=0;i<R-1;i++) a[i]=a[i+1];*

*R--;*

*return t;*

*}*

*int kirit\_right(int s){*

*if(isFull()){cout<<"\ndek to'ldi";n=R;return EXIT\_SUCCESS;} a[R]=s;R++;*

*}*

*int olish\_right(){*

*if(isEmpty()){cout<<"\ndek bo'sh";return EXIT\_SUCCESS;} R--;*

*return a[R];*

*}*

*int print(){*

*cout<<endl<<"dek ele-tlari=";*

*for(int i=0;i<R;i++)cout<<a[i]<<" "; cout<<endl;*

*}*

*int main(int argc, char \*argv[])*

*{ int n,s;cout<<"n="; cin>>n; for(int i=0;i<n;i++){*

*if(!isFull()){ cout<<"kirit=";cin>>s; if(i>=n/2) kirit\_right(s); else kirit\_left(s);}*

*else {cout<<"dek to'ldi\n";break;}*

*}*

*print();*

*int b[n/2],k=0,c[n/2],p=0;*

*while(!isEmpty()){*

*int q=olish\_left(); if(q%2==0) b[k++]=q; if(isEmpty()) break;*

*int p=olish\_right(); if(p%2==0) b[k++]=p;*

*}*

*int i=0; while(i<k){*

*kirit\_right(b[i]); i++;*

*}*

*print(); system("PAUSE"); return EXIT\_SUCCESS;*

*}*

**Dastur natijasi**

|  |  |
| --- | --- |
| kirit=1 |  |
| kirit=2 |
| kirit=3 |
| kirit=4 |
| kirit=5 |
| kirit=6 |
| kirit=7 |
| kirit=8 |
| dek ele-tlari=4 | 3 2 1 5 6 7 8 |
| dek ele-tlari=4 | 8 2 6 |

**Nazorat savollari**

* 1. Statik va yarimstatik Ma’lumotlar tuzilmasi nima va farqini tushuntiring?
  2. Navbat tuzilmasini tushuntiring.
  3. Yarimstatik Ma’lumotlar tuzilmasini dasturda e’lon qilishning qanday usullarini bilasiz?
  4. Stek tuzilmasini tushuntiring va misol keltiring.

Dek nima va navbat tuzilmasidan farqi nimada?

**Topshiriqlar**

1. Dekning har 2 ta elementidan keyin ularning yig„indisini joylang.
2. Dekning o„rtasidagi element yoki elementlarni o„chiring.
3. Dekning juft elementlari yig„indisini hisoblang.
4. Berilgan so’zning unli harflarini dekning chap tomonidan, undoshlarini o’ng tomondan kiriting.
5. Dekning toq elementlaridan navbat, juft elementlaridan stek hosil qiling.
6. Dekdagi manfiy sonlarni o„chiring.
7. Dekni o„rtasiga “dek” so’zini kiriting.

Har bir talaba 4 tadan misol ishlashi kerak!