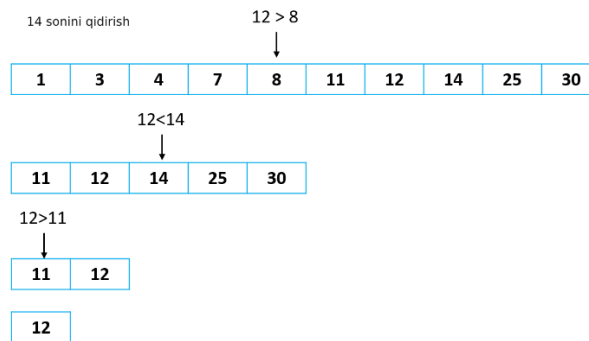


Mustaqil ishlash uchun masalalar

1. Binar qidiruv (binary search) yordamida tartiblangan massivdagi elementning qiymatning joylashgan o'rnini topish uchun dastur yozing. (Binar qidiruvning asosiy g'oyalaridan biri ketma-ket ikkiga bo'lishga asoslanadi, ya'ni berilgan x ni massivning o'rtadagi elementi bilan solishtiradi, agar katta bo'lsa oxiri va o'rtasi orasidagi massivni oladi, agar kichkina bo'lsa boshi va o'rtasi orasidagi massivni oladi, va har safar shu jarayon takrorlanib boradi toki x element solishtirilayotgan massivning elementga teng bo'lgunicha yoki massivning elementlari qolmaguncha.)



Dastur natijasi :

```
Massiv elementlari sonini kiriting : 7
7 ta elementni o'sish ta'rtibida kiriting: 1 2 3 4 5 6 7
Qidirmoqchi bo'lgan sonni kiriting : 2
2 qiymat 1 da topildi

Process returned 0 (0x0)   execution time : 13.066 s
Press any key to continue.
```

2. Interpolatsiya qidiruvi (Interpolation search) yordamida massiv ichidagi qiymatning joylashgan o'rnini topish uchun dastur yozing. (Interpolatsiya — bu butun soha va qidirilayotgan qiymatga o'xshash elementlar joylashishgan masofani hisoblash orqali qidiruv sohasini aniqlash usuli hisoblanadi. Bunga misol sifatida geometriyadagi o'xshash uchburchaklarni olish mumkin, bunda burchaklar qiymati bir xil, lekin

proportsiyasi har xil bo'ladi. Interpolyatsiya usulida ham aynan shunday printsiptan foydalaniladi.)

Dastur natijasi :

```
Berilgan massiv: 0 10 20 20 30 50 70 75 82 92 115 123 141 153 160 170
Qidirmoqchi bo'lgan sonni kiriting : 115
Element topildi : 10 chi indexta

Process returned 0 (0x0)   execution time : 1.208 s
Press any key to continue.
```

3. O'tish qidiruvi (Jump search) yordamida tartiblangan massivdagi qiymatning joylashgan o'rnini topish uchun dastur yozing.(Binar qidiruv singari, Jump Search ham tartiblangan massivlarni qidirish algoritmidir. Asosiy g'oya - belgilangan qadamlar bilan oldinga sakrab o'tish yoki barcha elementlarni qidirish o'rniga ba'zi elementlarni o'tkazib yuborish orqali kamroq elementlarni tekshirish.)

Dastur natijasi :

```
Berilgan massiv: 0 10 20 20 30 50 70 75 82 92 115 123 141 153 160 170
Qidirmoqchi bo'lgan qiymatni kiriting: 75
Element : 7 indexida topildi

Process returned 0 (0x0)   execution time : 4.252 s
Press any key to continue.
```

4. Chiziqli qidiruv (Linear search) yordamida massiv ichidagi qiymatning joylashgan o'rnini topish uchun dastur yozing.(Chiziqli qidiruv bir uchidan boshlanadigan va kerakli element topilgunga qadar ro'yxatning har bir elementi bo'ylab o'tadigan ketma-ket qidiruv algoritmi sifatida aniqlanadi, aks holda qidiruv ma'lumotlar to'plamining oxirigacha davom etadi.)

Dastur natijasi :

```
Berilgan massiv : 0 10 40 20 30 50 90 75 82 92 155 133 145 163 200 180
Qidirmoqchi bo'lgan qiymatni kiriting : 163
Element : 13 indexida joylashgan

Process returned 0 (0x0)   execution time : 2.022 s
Press any key to continue.
```

5. Uchlik qidiruv yordamida (Ternary search) massiv ichidagi qiymatning joylashgan o'rnini topish uchun dasturi yozing.(Uchlik qidiruv - massivdagi

elementni topish uchun ishlatilishi mumkin bo'lgan pasayish (doimiy) algoritmidir. Bu ikkilik qidiruvga o'xshaydi, bu erda biz massivni ikki qismga ajratamiz, lekin bu algoritmda biz berilgan massivni uch qismga ajratamiz va qaysi kalit (qidirilgan element) borligini aniqlaymiz.)

Dastur natijasi :

```
Berilgan massiv : 1 2 3 4 5 6 7 8 9 10
Qidirmoqchi bo'lgan qiymatni kiriting : 7
Element: 6indexida topildi

Process returned 0 (0x0)   execution time : 2.994 s
Press any key to continue.
```

6. Tanlovni saralash (Selection sort) algoritmidan foydalanib elementlar ro'yxatini kamayish tartibida saralash uchun dasturi yozing.(Tanlovni saralash oddiy va samarali tartiblash algoritmi bo'lib, ro'yxatning tartiblanmagan qismidan eng kichik (yoki eng katta) elementni qayta-qayta tanlab, uni ro'yxatning tartiblangan qismiga ko'chirish orqali ishlaydi.)

Dastur natijasi :

```
Massiv elementlari sonini kiriting : 10
Massiv elementlarini kiriting: 7 8 9 2 4 5 6 3 4 7 8
Saralangan massiv : 9 8 7 7 6 5 4 4 3 2

Process returned 0 (0x0)   execution time : 11.830 s
Press any key to continue.
```

7. Pufakchali tartiblash (Bubble sort) algoritmidan foydalanib elementlar ro'yxatini tartiblash uchun dastur yozing.(Bubble Sort - bu eng oddiy tartiblash algoritmi bo'lib, agar ular noto'g'ri tartibda bo'lsa, ulashgan elementlarni qayta-qayta almashtirish orqali ishlaydi. Ushbu algoritm katta ma'lumotlar to'plamlari uchun mos emas, chunki uning o'rtacha va eng yomon vaqt sarflashi ancha yuqori.)

Dastur natijasi :

```
Berilgan massiv : 15 56 12 -21 1 659 3 83 51 3 135 0
Tartiblangan massiv : -21 0 1 3 3 12 15 51 56 83 135 659

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```

8. Tartiblash (insertion sort algorithm.) algoritmidan foydalanib elementlar ro'yxatini tartiblash uchun dastur yozing.(Insertion sort algorithm - bu oddiy tartiblash algoritmi bo'lib, u sizning qo'lingizda o'yin kartalarini saralash usuliga o'xshash ishlaydi. Massiv deyarli tartiblangan va tartiblanmagan qismga bo'lingan. Saralanmagan qismdan qiymatlar tanlanadi va tartiblangan qismning to'g'ri joyiga joylashtiriladi.)

Dastur natijasi :

```
Massiv o'lchamini kiriting : 10
10 ta elementlarini kiriting : 4 5 8 3 12 65 41 21 87 52 33
tartiblangan massiv : 3 4 5 8 12 21 41 52 65 87

Process returned 0 (0x0)   execution time : 17.603 s
Press any key to continue.
```

9. Birlashtirish tartiblash (merge sort) algoritmidan foydalangan holda massiv elementlar ro'yxatini saralash uchun dastur yozing.(Birlashtirish saralash (merge sort) massivni kichikroq kichik massivlarga bo'lish, har bir kichik massivni saralash va so'ng saralangan pastki massivlarni yana birlashtirib, yakuniy tartiblangan massivni hosil qilish orqali ishlaydigan tartiblash algoritmi sifatida aniqlanadi.)

Dastur natijasi :

```
Berilgan massiv : 125 181 130 25 61 887 45 21 32 75
Tartiblangan massiv : 21 25 32 45 61 75 125 130 181 887

Process returned 0 (0x0)   execution time : 0.030 s
Press any key to continue.
```

10. MAX heap algoritmidan foydalanib raqamlarni tartiblash uchun dastur yozing.(Max-Heap har bir ichki tugun (node) o'z vorislariga teng yoki kattaroq bo'lgan yig'ma ma'lumotlar strukturasi turi sifatida aniqlanadi.)

Dastur natijasi :

```
Massiv o'lchamini kiriting : 10
Massiv elementlarini kiriting : 4 2 8 32 12 41 54 65 32 10
Tariblangan massiv : 2 4 8 10 12 32 32 41 54 65

Process returned 0 (0x0)   execution time : 13.350 s
Press any key to continue.
```

Mavzu yuzasidan testlar

1. Berilgan argumentga mos keluvchi ma'lumotlarni massiv ichidan toppish
 - a) Qidiruv*
 - b) Saralash
 - c) Algoritmash
 - d) Uslubiyot
2. Chiziqli qidiruv g'oyasi nimadan iborat
 - a) har bir element ketma-ket ko'rib chiqiladi*
 - b) elementlar ketma-ket jadval o'rtasidan boshlab ko'rib chiqiladi
 - c) elementlarni ko'rib chiqish ketma-ket ravishda boshidan oxirigacha va aksincha, 2 ta element tashlab qaraladi
 - d) binar daraxt barcha tugunlari ko'rib chiqiladi
3. Noyob kalit nima
 - a) agar jadvalda kaliti mazkur kalitga teng ma'lumot yagona bo'lsa*
 - b) agar ikkita ma'lumot qiymatlari yig'indisi kalitga teng bo'lsa
 - c) agar jadvalda bunday kalitli element mavjud bo'lmasa
 - d) agar ikkita ma'lumot qiymatlari farqi kalitga teng bo'lsa
4. Katta O notasiyada belgilangan chiziqli qidiruv samaradorligini ko'rsating
 - a) $O(n)^*$
 - b) $O(\log_2 n)$
 - c) $O(1)$
 - d) $O(\log n)$

5. Katta O notasiyada belgilangan binar qidiruv samaradorligini ko'rsating
 - a) $O(n)$
 - b) $O(\log_2 n)^*$
 - c) $O(1)$
 - d) $O(\log n)$
6. Binar qidiruv algoritmi tartibi qanday
 - a) Chiziqli
 - b) Konstantali
 - c) Logarifmik*
 - d) Eksponensial
7. Ketma-ket qidiruv algoritmi tartibi qanday
 - a) Chiziqli*
 - b) Konstantali
 - c) Logarifmik
 - d) Eksponensial
8. Binar qidiruv – bu ...
 - a) Ma'lumotlar butun jadval bo'yicha operativ xotirada kichik adresdan boshlab, to katta adresgacha ketma-ket qarab chiqiladi*
 - b) Berilgan massiv o'rtasidagi element olinadi, ya'ni, va u qidiruv argumenti bilan taqqoslanadi. Topilmasa chegaralar mos ravishda o'zgartiriladi
 - c) Funksiya yordamida xesh-jadval to'ldiriladi va undan qidiriladi
 - d) Indekslar jadvalidan gurux topiladi, va unda ko'rsatilgan mos chegaralarda chiziqli algoritmi oshiriladi
9. Tezkor xotirada bajariladigan saralash qanday ataladi
 - a) to'liq saralash
 - b) adreslar jadvalini saralash
 - c) ichki saralash*

d) adreslar jadvalini saralash

10. Qidiruvni vazifasi nimadan iborat

a) berilgan argumentga mos keluvchi ma'lumotlarni massiv ichidan toppish*

b) massivda ma'lumot yo'qligini aniqlash

c) ma'lumotlar yordamida argumentni toppish

d) ma'lumot yordamida eng kichik elementni topish