

5-Ma'ruza. C++ da satrli kattaliklar v ular bilan ishlash

Ma'ruza rejasi:

5.1 Belgili axborot va satrlar;

5.2 So'zlar massivlari;

5.3 Izlash va tartiblash.

Kalit so'zlar: *delete, masofa keltirish, delete[], new, indeks, this, indeksirlash, [] bo'sh xotira, void*, konteyner, ro'yxat, manzil, nolinchi ko'rchsatkich, tugun, adres olish &, bo'shatish, ko'rsatkich, virtual destruktor, xotira, xotira chiqishi, destruktor, toifani o'zlashtirish, resurslar chiqishi, a'zo destruktori.*

5.1. Belgili axborot va satrlar

Satrlar. C da belgili ma'lumotlar uchun char turi qabul qilingan. Belgili axborotni taqdim etishda belgilar, simvolli o'zgaruvchilar va matnli konstantalar qabul qilingan.

Misollar:

```
const char c = 'c';
```

```
char a,b;
```

S dagi satr - bu nul-belgi - \0 (nul-terminator)- bilan tugallanuvchi belgilar massivi. Nul-terminatorning holatiga qarab satrning amaldagi uzunligi aniqlanadi. Bunday massivdagi elementlar soni, satr tasviriga qaraganda, bittaga ko'p.

Simvolli massivlar quyidagicha inisializasiya qilinadi:

`char capital[] = "TASHKENT";` Bu holda avtomatik ravishda massiv elementlari soni aniqlanadi va massiv oxiriga satr ko'chirish `'\0'` simvoli qo'shiladi.

Yuqoridagi inisializasiyani quyidagicha amalga oshirish mumkin:

```
char capital[] = {'T','A','S','H','K','E','N','T','\0'};
```

Bu holda so'z oxirida `'\0'` simvoli aniq ko'rsatilishi shart.

Qiymat berish operatori yordamida satrga qiymat berish mumkin emas. Satrni massivga yoki kiritish paytida yoki nomlantirish yordamida joylashtirish mumkin.

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
char s1[10] = "string1";
int k = sizeof(s1);
printf("\n%s %d",s1,k);
char s2[] = "string2";
k = sizeof(s2);
printf("\n%s %d",s2,k);
char s3[] = {'s','t','r','i','n','g','\0'};
k = sizeof(s3);
printf("\n%s %d",s3,k);
char *s4 = "string4";//satr ko'rsatkichi, uni o'zgartirib bo'lmaydi
k = sizeof(s4);
printf("\n%s %d",s4,k);
system("pause");
return 0;
}

```

Natija:

string1 10

string2 8

string3 8

string4 4

Keyingi misolda kiritilgan so'zdan berilgan harfni olib tashlash dasturi berilgan.

```

#include <stdio.h>
int main()
{
char s[100];
scanf("%s",&s);
int i, j;
for ( i = j = 0; s[i] != '\0'; i++)
if ( s[i] != 'c' )
s[j++] = s[i];

```

```

s[j] = '\0';
printf("%s",s);
return 0;
}

```

Xar safar 's' dan farqli simvol uchraganda, u j pozitsiyaga yoziladi va faqat shundan so'ng j ning qiymati 1 ga oshadi. Bu quyidagi yozuvga ekvivalent:

```

if ( s[i] != c )
s[j] = s[i];
j++;

```

Funksiyalar va satrlar. Funksiyalarda satrlar ishlatilganda ularning chegarasini ko'rsatish shart emas. Satrlarning uzunligini hisoblash len funksiyasini quyidagicha ta'riflash mumkin:

```

int len(char c[])
{
    int m = 0;
    for(m = 0; c[m] != '\0'; m++);
    return m;
};

```

Shu funksiyadan foydalanilgan dasturni keltiramiz:

#include <stdio.h>	int main()
int len(char c[])	{
{	char e[] = "Pro Tempore!";
int m = 0;	printf("\n%d", len(e));
while(c[m++]);	return 0;
return m-1;	}
};	

Bu funksiyaning standart varianti strlen deb ataladi va bu funksiyadan foydalanish uchun string.h sarlavha faylidan foydalanish lozim.

Satrdan nusxa olish funksiyasi strcpy ni C tilida quyidagicha ta'riflash mumkin:

#include <stdio.h>	s1[i] = s2[i];
void strlen(char s1[], char s2[])	}
{	
int i = 0;	int main()
while(s2[i] != '\0') s1[i++] = s2[i];	{

char s1[] = "aaa";	reverse(char s[])
char s2[] = "ddd";	{
strcpy(s1,s2);	int c, i, j;
printf("%s",s1);	for(i = 0, j = strlen(s) - 1; i < j; i++, j-
return 0;	-)
}	c = s[i];
Natija:	s[i] = s[j];
ddd	s[j] = c;
	}

Berilgan satrni teskariga aylantiruvchi

funksiya:

Keyingi misolimizda T qatorni S qator oxiriga ulovchi STRCAT(S,T) funksiyasini ko'rib chiqamiz:

strcat(char s[], t[])	while (s[i] != '\0')
{	i++;
int i, j;	while((s[i++] = t[j++]) != '\0')
i = j = 0;	}

5.2. So'zlar massivlari

So'zlar massivini kiritish. C tilida so'zlar massivlari ikki o'lchovli simvolli massivlar sifatida ta'riflanadi. Misol uchun:

```
char name[4][5].
```

Bu ta'rif yordamida har biri 5 ta harfdan iborat bo'lgan 4 ta so'zli massiv kiritiladi. So'zlar massivlari quyidagicha inisializasiya qilinishi mumkin:

```
char Name[3][8] = { "Anvar", "Mirkomil", "Yusuf"}.
```

Bu ta'rifda har bir so'z uchun xotiradan 8 bayt joy ajratiladi va har bir so'z oxiriga ' \0' belgisi kuyiladi.

So'zlar massivlari inisializasiya qilinganda so'zlar soni ko'rsatilmasligi mumkin. Bu holda so'zlar soni avtomatik aniqlanadi:

```
char comp[][9] = { "kompyuter", "printer", "kartridj"}.
```

Quyidagi dasturda berilgan harf bilan boshlanuvchi so'zlar ro'yxati bosib chiqariladi:

```

#include <stdio.h>
int main()
{
    char a[10][10];
    char c = 'a';

    int i;
    for (i = 0; i < 3; i++) scanf("%s", &a[i]);
    for (i = 0; i < 3; i++)
        if (a[i][0] == c) printf("\n%s", a[i]);
    return 0;
}

```

Quyidagi dasturda fan nomi, talabalar ro'yxati va ularning baholari kiritiladi. Dastur bajarilganda ikki olgan talabalar ro'yxati bosib chiqariladi:

```

#include <stdio.h>
int main()
{
    char a[10][10];
    char s[10];
    int k[10];
    scanf("%s", &s);
    for (int i = 0; i < 3; i++)
        {
            scanf("%s", &a[i]);
            scanf("%d", &k[i]);
        };
    for (int i = 0; i < 3; i++)
        if (k[i] == 2) printf("%s\n", a[i]);
    return 0;
}

```

Funksiyalar va so'zlar massivlari. Satrli massivlar funksiya argumenti sifatida ishlatilganda satrlarning umumiy uzunligi aniq ko'rsatilishi shart.

Misol tariqasida ixtiyoriy sondagi satrlar massivini alfavit bo'yicha tartiblash funksiyasidan foydalanilgan dasturni ko'rib chiqamiz:

```

#include <stdio.h>
#define m 10
void sort(int n, char a[][m])
{
    char c;
    int i, j, l;
    for (i = 0; i < n; i++)
        for (j = i + 1; j < m; j++)
            if (a[i][0] < a[j][0])
                for (l = 0; l < m; l++)
                {
                    c = a[i][l];
                    a[i][l] = a[j][l];
                    a[j][l] = c;
                }
}
int main()
{
    char aa[][m] = {"Alimov", "Dadashev", "Boboev"};
    sort(3, aa);
    for (int i = 0; i < 3; i++)
        printf("%s\n", aa[i]);
    return 0;
}

```

5.3. Izlash va tartiblash

Ikkiga bo'lib izlash. Quyidagi dasturda tartiblangan massivda ikkiga bo'lib kalit sonni izlash algoritmi asosida tuzilgan funksiyadan foydalanish keltirilgan:

```
#include<stdio.h>
#include<conio.h>
int bsearch(int a[],int key,int n)
{
    int m1,m2,m;
    m1 = 0;m2 = n-1;
    while(m1<= m2)
    {
        m = (m2+m1)/2;
        if (a[m] == key) return m;
        if (a[m]>key) m2 = --m;
        if (a[m]<key) m1 = ++m;
    }
    return -1;
};

int main(int argc, char* argv[])
{
    int m;
    int a[] = {5,6,9,11};
    m = bsearch(a,7,4);
    printf("%d",m);
    getch();
    return 0;
}
```

Keyingi misolda shu funksiya satrlar uchun varianti keltirilgan:

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
#define size 5
int strbsearch(char a[][size],char
key[],int n)
{
    int m1,m2,m,pr;
    m1 = 0;m2 = n-1;
    while(m1<= m2)
    {
        m = (m2+m1)/2;
        pr = strcmp(a[m],key);
        if (pr == 0) return m;
        if (pr == -1) m2 = --m;
        if (pr == 1) m1 = ++m;
    }
    return -1;
};

int main(int argc, char* argv[])
{
    int m;
    char a[][size] =
    {"aaa","aab","aac","aad"};
    m = strbsearch(a,"aab",4);
    printf("%d",m);
    getch();
    return 0;
}
```

Tezkor tartiblash. Quyidagi dasturda tezkor tartiblash algoritmiga asoslangan funksiyadan foydalanilgan. Algoritm mohiyati shundan iboratki, avval yetakchi element tanlanadi. Funksiyada yetakchi element sifatida boshlang'ich element tanlanadi. Shundan so'ng massiv ikki

qismga ajratiladi. Yetakchi elementdan kichik elementlar past qismga, katta elementlar yuqori qismga to'planadi. Shundan so'ng rekursiya asosida algoritm ikkala qismga alohida qo'llanadi.

```
#include<stdio.h>
#include<conio.h>
int a[] = {5,4};
void sqsort(int k1, int k2)
{
if(k1 < k2){
int i, j, k;
i = k1; j = k2;
while(i < j)
{
if (a[k1] > a[i]) {i++; continue;}
if (a[k1] < a[j]) {j--; continue;}
k = a[j]; a[j] = a[i]; a[i] = k;
}
}
k = a[k1]; a[k1] = a[i]; a[i] = k;
sqsort(k1, i);
sqsort(i+1, k2);
};
}
int main(int argc, char* argv[])
{
int i;
sqsort(0, 1);
for(i = 0; i < 2; i++) printf("%d ",
a[i]);
getch();
return 0;
}
```

Nazorat savollari

- 1.Satr simvolli massivdan qanday farq qiladi?
- 2.Bir o'lchovli massivlarni inisializasiya qilish usullarini ko'rsating.
- 3.Ko'p o'lchovli massiv ta'rifi xususiyatlarini keltiring.
- 4.Ko'p o'lchovli masivlar inisializasiyasi xususiyatlari.
- 5.Satrlarni inisializasiya qilish usullarini ko'rsating.
- 6.So'zlar massivi qanday kiritiladi?
- 7.Qanday qilib bir o'lchovli massivlar formal parametrlar sifatida ishlatilishi mumkin?
- 8.Qanday qilib ko'p o'lchovli massivlar formal parametrlar sifatida ishlatilishi mumkin?
- 9.Satr ta'riflash usullari.
10. Satrlar funksiya parametri sifatida.