

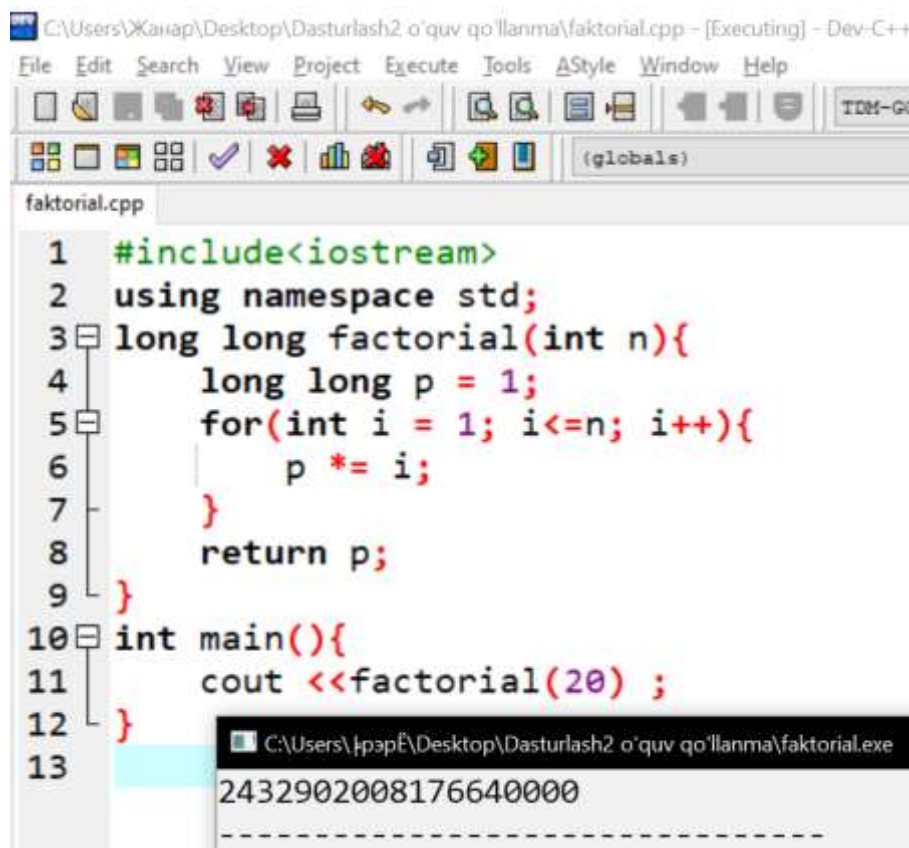
DASTURLASHDA JUDA HAM KATTA SONLAR BILAN ISHLASH

Yusupova Janar,² Anvarova Dilrabo,³ Omonov Sardorbek,⁴ Raximov Komron

¹ Muhammada al-Xorazmiy nomidagi TATU Urganch filiali o'qituvchisi

^{2,3,4} Muhammada al-Xorazmiy nomidagi TATU Urganch filiali talabasi

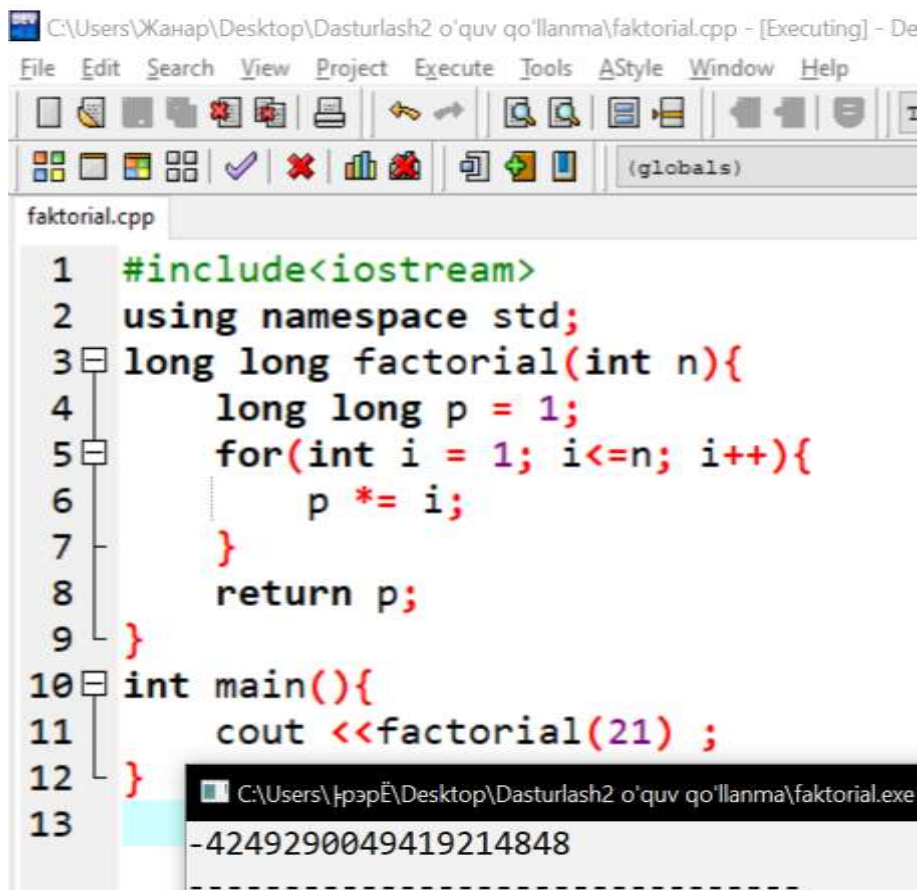
C va C++ dasturlash tillari ikkita asosiy butun son turiga ega: *int* va *long*, odatda mos ravishda 32 va 64 bayt hajmga ega. Ya'ni, *int* turi -2147483648 dan 2147483647 gacha, *long long* turi esa -9223372036854775808 dan 9223372036854775807 gacha bo'lgan sonlarni o'z ichiga olishi mumkin. Ko'p xonali sonlarni talab qilmaydigan kundalik hisoblar uchun bu turlar yetarli bo'ladi. *long long* turidan foydalanib, yigirma sonining faktorialini ($20! = 2432902008176640000$) hisoblash oson (1-rasm), lekin yigirma bir sonining faktoriali ($21! = 1090942171709440000$) endi unga mos kelmaydi. Bunda berilgan son turning chegaraviy qiymatidan oshib ketadi va dastur noto'g'ri natija qaytaradi (2-rasm).



```
C:\Users\Xavap\Desktop\Dasturlash2 o'quv qo'llanma\faktorial.cpp - [Executing] - Dev-C++
File Edit Search View Project Execute Tools AStyle Window Help
(fglobals)
faktorial.cpp
1 #include<iostream>
2 using namespace std;
3 long long factorial(int n){
4     long long p = 1;
5     for(int i = 1; i<=n; i++){
6         p *= i;
7     }
8     return p;
9 }
10 int main(){
11     cout <<factorial(20) ;
12 }
13
```

C:\Users\Xavap\Desktop\Dasturlash2 o'quv qo'llanma\faktorial.exe
2432902008176640000

1-rasm. Sonning faktorialini hisoblash dasturi



```
1 #include<iostream>
2 using namespace std;
3 long long factorial(int n){
4     long long p = 1;
5     for(int i = 1; i<=n; i++){
6         p *= i;
7     }
8     return p;
9 }
10 int main(){
11     cout <<factorial(21) ;
12 }
13
```

C:\Users\pəpƏ\Desktop\Dasturlash2 o'quv qo'llanma\faktorial.exe
-4249290049419214848

2-rasm. Hisoblash natijasi berilgan turning chegaraviy qiymatidan oshib ketishi

Agar ma'lumotlar turi chegarasidan oshib ketadigan qiymatlarni olish kerak bo'lsa, unda nima qilish kerak? Bunday holatlarda qurilmada emas, balki dasturda amalga oshiriladigan katta sonlar arifmetikasi yordamga keladi. Bu arifmetikada sonlar boshqa shaklda, masalan, raqamlar massivi yoki satrda saqlanadi va sonlar ustida amallar raqamlarning o'ziga qo'llaniladigan algoritmlar yordamida amalga oshiriladi[1].

Katta sonlar arifmetikasi bizga ixtiyoriy ravishda katta sonlarni hisoblash imkonini beradi (raqamlarni satrda (std::string) saqlashda 9223372036854775807 xonali sonlar bilan ishlashimiz mumkin). Biroq, juda muhim kamchilik mavjud: an'anaviy ma'lumotlar turlari bilan solishtirganda, bu usul juda ko'p vaqt talab qiladi, chunki barcha operatsiyalar barcha raqamlar uchun dasturiy ravishda amalga oshiriladi.

Katta sonlar bilan ishlaganda, yondashuvni boshqacha amalga oshirish mumkin, masalan oqimdan kiruvchi raqamlarni massiv elementlari deb qarab uni ustida mustaqil arifmetik amallarni bajarish mumkin[2].

Har bir arifmetik amalni oldindan C++ dasturlash tilida amalga oshirish dastur fragmentini yozib, alohida ko'rib chiqish mumkin. Avvalo, cheksiz katta sonni faqat dinamik massiv sifatida ifodalash mumkinligini tushunish kerak. Lekin raqamlari dinamik massiv sifatida taqdim etilsa ham, ba'zi cheklovlar bo'ladi. Masalan, bunday sonning (massivning) uzunligi kompyuter xotirasi miqdori bilan chegaralangan bo'ladi. Shuni ham tushunish kerakki, qo'shish va ayirish amallaridan foydalanishda natija operandlarga qaraganda kompyuter xotirasida ko'proq joy egallaydi.

Ko'p sonli hisoblashlar standart ma'lumot turlarining cheklanishidan tashqariga chiqish, cheksiz katta sonlar bilan ishlash, hajmi faqat mashinaning hisoblash kuchi bilan cheklanadi. Bunga qanday erishish mumkin? Eng yaxshi yo'li massivdan foydalanish va standart tipga mansub bo'lgan sonlarni izchil massiv uchun maxsus tarzda aylantirish. Misol uchun, 123456789123456789 sonini qanday saqlash mumkin, u int tipiga mos kelmaydi. Uni joylash uchun, int tipidagi massiv olish mumkin, $arr[0] = 123456$, $arr[1] = 789123$, $arr[2] = 456789$, bunday yozishni maxsus usullari deb atash mumkin. Shuningdek, to'g'ri kiritilsa, qo'shish, ayirish, ko'paytirish va bo'lish amallarini yozish mumkin.

Katta sonlar arifmetikasini amalga oshirish algoritmlash nuqtai nazaridan ham, dasturlash amaliyoti nuqtai nazaridan ham juda qiziqarli mashg'ulot hisoblanadi.

Katta sonlar arifmetikasi - bu standart ma'lumotlar turlaridan ancha katta bo'lgan sonlar bilan ishlash imkonini beruvchi dasturiy vositalar (ma'lumotlar tuzilmalari va algoritmlar) to'plami. Bunda asosiy g'oya shundan iboratki, son uning raqamlari qatori sifatida saqlanadi. Bunday arifmetikada sonlar ustida amallar qo'shish, ayirish, ko'paytirish, bo'lish amallari "maktab" algoritmlari yordamida bajariladi. Biroq, ular uchun tez ko'paytirish algoritmlari hisoblangan *Fast Fourier Transform* va *Karatsuba* algoritmlari ham qo'llaniladi[3].

ADABIYOTLAR

1. Turaboyevich A. O. et al. DASTURLASHNI O 'RGANUVCHILAR UCHUN YARATILGAN "ONLINE HAKAM" TIZIMLARI TAHLILI //Komputer texnologiyalari. – 2022. – T. 1. – №. 10.
2. Yusupova J., Choponov O., Allayarov S. PARALLEL DATA TESTING ON" ONLINE HAKAM" SYSTEMS FOR PROGRAMMING STUDENTS //Science and innovation. – 2023. – T. 2. – №. A6. – C. 331-334.
3. Allamov O. et al. Analysis of Parallel Computing Methods and Algorithms //2023 IEEE XVI International Scientific and Technical Conference Actual Problems of Electronic Instrument Engineering (APEIE). – IEEE, 2023. – C. 1710-1713.