1. **Алгоритм гэж юу вэ ?**

Алгоритмын тухай ойлголт нэгэн төрлийн бодлогуудыг бодох ерөнхий арга олох гэсэн оролдлоготой холбоотойгоор үүссэн. Алгоритм гэдэг нэр нь персийн математикч Abu Ja’far Mohammed ibn Musa al Khowarizmi –н нэрнээс гаралтай. Аливаа бодлогыг бодох (ажлыг гүйцэтгэх) үйлдлүүдийн төгсгөлөг дарааллыг алгоритм гэнэ. Програмыг компьютерт шууд бичиж оруулах ёстой юу ? Тэгж болно. Гэхдээ програмчлалд анхлан суралцаж буй хүний хувьд эхлээд бичих гэж буй програмынхаа загвар буюу бүдүүвчийг зохиогоод дараа нь уг бүдүүвчийнхээ дагуу програмаа бичих нь илүү үр дүнтэй юм. Ийм загвар буюу төлөвлөгөөг алгоритм гэж бас хэлж болно. Алгоритм нь дараах шинжүүдтэй. Үүнд:

* ***Үр дүнтэй байх***
* ***Үнэн зөв байх***
* ***Төгсгөлөг байх***
* ***Түгээмэл байх*** Өөрөөр хэлбэл өгөгдлөөрөө ялгаатай ижил төрлийн бодлогыг бодох алгоритм ижил байна.

Алгоритмд хэмжигдэхүүнүүдийг өөр хооронд ялгах, тэдгээр дээр хийх үйлдлийг бичихийн тулд үг үсгээр тэмдэглэдэг. Энэ тэмдэглэгээг хэмжигдэхүүний нэр гэнэ. Нэр нь заавал үсгээр эхэлсэн байх ба дурын тооны үсэг, цифрээс тогтоно. Хэмжигдэхүүний нэр нь уг хэмжигдэхүүний утга хадгалагдаж байгаа санах ойн үүрийн хаягийг төлөөлнө. Алгоритмын явцад хэмжигдэхүүний утга өөрчлөгдөхгүй байвал түүнийг тогтмол хэмжигдэхүүн гэнэ. Эсрэг тохиолдолд хувьсагч гэнэ. Хэмжигдэхүүнийг үйлдлийн тэмдгээр холбож илэрхийлэл үүсгэнэ.

**Үйлдлүүд**

1. **Арифметик үйлдэл** Нэмэх(+), хасах(-), үржих (\*), хуваах(/), бүхэл тоог бүхэл тоонд хуваахад гарах үлдэгдэл олох (%)
2. **Логик үйлдэл** Зөвхөн үнэн эсвэл худал утга авдаг хэмжигдэхүүнийг логик хэмжигдхүүн гэнэ. Логик хэмжигдэхүүн дээр логик нэмэх (or), логик үржих(and), үгүйсгэл (not) үйлдлүүдийг хийдэг. Цаашид үнэн утгыг 1, худал утгыг 0 гэж тэмдэглэе. Логик хэмжигдэхүүн дээрх үйлдлийг дараах хүснэгтээр үзүүлье.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | X and Y | X or Y | not X |
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 |

1. **Жиших үйлдэл** (>=,<=,<,>,≠,=)
2. Функц хэрэглэх sin(x), cos(x), tg(x), ctg(x), |x|, ln(x), , arctg(x), [x] - x тооны бүхэл хэсэг, {x} – x тооны бутархай хэсэг.

Алгоритм нь дараах алхмуудаас тогтоно. үүнд:

* **Оруулах алхам**: Хувьсагчдын анхны утгыг оруулж өгнө. Жишээ нь ax2+bx+c=0 тэгшитгэлийг бодохын тулд a,b,c хувьсагчдын утгуудыг оруулж өгнө.
* **Утга олгох алхам**: Тодорхой томъёогоор өгөгдсөн илэрхийллийн утгыг бодох, гарсан үр дүнг ямар нэг хувьсагчийн утга болгон хадгалах үйлдлийг утга олгох үйлдэл гэнэ. Энэ үйлдлийг ***хувьсагч=илэрхийлэл*** гэж гүйцэтгэнэ.
* **Гаргах алхам:** Алгоритмын үр дүнг мэдээлэх шаардлагатай байдаг.
* **Нөхцөл шалгах алхам:** Алгоритмд тодорхой нөхцөл шалгаж нөхцөл биелэх эсэхээс хамаарч алгоритмыг ялгаатай замаар үргэлжлүүлэх шаардлага гардаг.

1. **Алгоритмыг дүрслэх хэлбэрүүд**

***Алгоритмыг үгээр бичих, блок схем ашиглах, псевдо кодоор бичих гэх мэт хэд хэдэн аргаар тэмдэглэдэг.***

Алгоритмын алхмуудыг псевдо кодоор хэрэхн тэмдэглэхиийг дараах хүснэгтэд харуулав.

|  |  |  |
| --- | --- | --- |
| Алхам | псевдо кодын бичлэг | Жишээ |
| 1. Оруулах алхам 2. Утга олгох алхам 3. Гаргах алхам 4. Нөхцөл шалгах     5. Эхлэл ,төгсгөл  6. Тайлбар | Read(оруулах утгууд); Хувьсагч=илэр-хийлэл Write(гаргах утга); **if**(нөхцөл) **{** нөхцөл үнэн   үеийн үйлдлүүд **}**  **else {** нөхцөл худал  үеийн үйлдлүүд **}**  **{** -алгоритмын эхлэл **}**- алгоритмын төгсгөл  **//** тайлбар | Read(a,b); a=a+b; |

òàéëáàð

i, j

**Жишээ 1** Өгөгдсөн 3 оронтой тооны цифрүүдийн нийлбэрийг ол.

**{** **// өгөгдсөн тоог оруулах**

read(n);

// a зуутын орны цифр, b аравтын орны цифр, c нэгжийн орны цифр

a=[n/100];

b=[(n%100) /10];

c=n%10;

write(a+b+c);

**}**

1. **Алгоритмын үндсэн бүтцүүд**

Элементүүд нийлж тодорхой бүтцийг үүсгэдэг. Эдгээр бүтцийг алгоритмын үндсэн бүтэц (АҮБ) хэмээн нэрлэнэ. Ингэснээр алгоритмыг зөвхөн эдгээр бүтцээс тогтсон мэтээр авч үзэх боломжтой. АҮБ –ийн гол онцлог бол тэдгээр нь зөвхөн нэг оролт, нэг гаралттай байна. 3 төрлийн АҮБ байдаг. Энэ нь Процесс, Салаалалт, Давталт юм. Салаалалт бүтэц нь гүйцэд ба гүйцэд бус гэсэн 2 янз байдаг. Давталт бүтэц нь while (байхад), until (хүртэл), параметрт гэсэн 3 хэлбэртэй.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Процесс** | **Салаалалт** | | | |
| Гүйцэд | | Гүйцэд бус | |
| **Үйлдэл**  **Үйлдэл**  **Урсгал 1**  **Урсгал 2**  **нөхцөл**  **Урсгал**  **нөхцөл**  **Нөхцөл**  **Урсгал**  **Нөхцөл**  **Урсгал**  **ДП=АУ, ЭУ, АЛХАМ**  **Урсгал** |  | |  | |
| **Давталт** | | | | |
| While | | Until | | Параметрт |
| худал  үнэн | | үнэн     худал | |  |

1. **Алгоритмын төрлүүд**

Алгоритмыг алхмуудын биелэгдэх дэс дарааллаар нь шугаман, салаалсан, давталттай гэж ангилна.

**Шугаман алгоритм**

Алхмууд нь дэс дарааллаараа зөвхөн ганц удаа биелэгдэж байвал шугаман алгоритм гэнэ.

**Жишээ 2**  Өгөгдсөн R радиустай тойргийн урт ба талбайг ол.

L=2πR, S=πR2

**{** Pi=3.141592;

read(R);

L=Pi\*R;

S=L\*R;

L=2\*L;

write(S,L);

**}**

# **Салаалсан алгоритм**

Алгоритмын зарим алхам биелэгдэх эсэх нь ямар нэг нөхцлөөс хамаарч ялгаатай замаар биелэгдэж байвал тийм алгоритмыг салаалсан алгоритм гэнэ. Дараах 2 төрлийн салаалсан алгоритм байна.

**a. if**(нөхцөл) **{** нөхцөл үнэн

үеийн үйлдлүүд

**}**

**б.** **if**(нөхцөл) **{** нөхцөл үнэн

үеийн үйлдлүүд

**}**

**else{** нөхцөл худал

үеийн үйлдлүүд

**}**

**Жишээ 3** x ба y бодит тоонууд өгөгджээ. z-ийн утгыг дараах томъёогоор ол.



**{** read(x,y);

if(x>y) z=x-y;

else z=y-x+1;

write(z);

**}**

**Жишээ 4** ax2+bx+c=0 тэгшитгэлийг бод.

1. a≠0 үед
2. D>0 бол 2 шийдтэй.
3. D=0 бол x=-b/2a гэсэн давхар шийдтэй.
4. D<0 бол бодит шийдгүй.
5. a=0 үед
6. b≠0 бол x=-c/b гэсэн 1 шийдтэй.
7. b=0 үед
8. c≠0 бол шийдгүй.
9. c=0 бол төгсгөлгүй олон шийдтэй.

**{** read(a,b,c);  
 if(a=0)   
 if(b=0)  
 if(c=0) write(“адилтгал”);  
 else write(“шийдгүй”)д  
 else write(-c/b);  
 else **{** d=b\*b-4\*a\*c;  
 if(d<0) write(“бодит шийдгүй”);  
 else if(d=0) write(x1=-b/(2\*a));  
 else **{** d=sqrt(d);  
 x1=(-b+d)/(2\*a);  
 x2=(-b-d)/(2\*a);  
 write(x1,x2);  
 **}** **}  
}**

**Давталттай алгоритм**

Алгоритмын нэг юмуу хэсэг алхмууд хэд хэдэн удаа давтагдан биелэгдэж байвал уг алгоритмыг давталттай алгоритм гэнэ. Давталтын алхмуудыг давталтын бие буюу циклийн бие гэнэ. Давталттай алгоритмыг тоолуурт давталт (параметрт), нөхцөлт давталт гэж ангилна. ßмар нэг нөхцлийг биелэхгүй (эсвэл биелэх) болтол бүлэг алхмуудыг давтаж байвал нөхцөлт давталт гэнэ. Нөхцөлт давталтыг өмнөх нөхцөлтэй, дараах нөхцөлтэй гэж ангилна. Нөхцөлт давталтыг псевдо кодоор дараах хэлбэрээр тэмдэглэнэ.

while(нөхцөл){ do{  
 давталтын бие давталтын бие   
 } }while(нөхцөл);  
 a***. Өмнөх нөхцөлтэй б. Дараах нөхцөлтэй  
 давталт давталт***

**Жишээ 1.** m (m>1)- бүхэл тоо өгөгджээ. 4k<m тэнцэл биш биелэх к-ийн хамгийн их бүхэл утгыг ол.

|  |  |
| --- | --- |
| // өмнөх нөхцөлтэй давталтаар бодъё.  **{** read(m);  p=1; k=0;  while(p<m)**{** k=k+1;  p=4\*p;  **}**  write(k-1);  **}** | // дараах нөхцөлтэй давталтаар бодъё.  **{** read(m);  p=1; k=0;  do**{** k=k+1;  p=4\*p;  **}**while(p<m);  write(k-1);  **}** |

**Жишээ 2.**  Өгөгдсөн n натурал тооны орон ба цифрүүдийн нийлбэрийг ол.

Бодолт. S-ээр цифрүүдийн нийлбэрийг, k-аар оронгийн тоог, m-ээр n тоог 10-т хуваахад гарах үлдэгдлийг тус тус тэмдэглэе.

|  |  |
| --- | --- |
| // өмнөх нөхцөлтэй давталтаар бодъё.  **{**read(n);  s=0; k=0;  while(n>0)**{** m=n%10;  s=s+m;  k=k+1;  **}**  write(s,k);  **}** | // дараах нөхцөлтэй давталтаар бодъё.  **{**read(n);  s=0; k=0;  do**{** m=n%10;  s=s+m;  k=k+1;  **}**while(n>0);  write(s,k);  **}** |

Харин давталтын тоо тодорхой бол тоолуурт давталт гэнэ. Давталтыг тоолж байгаа хувьсагчийг тоолуур гэнэ. Тоолуурт давталтыг зохиохдоо :

* тоолуурын эхний утгыг олгоно.
* Тоолуурын утга эцсийн утгадаа хүрээгүй бол давталтын биеийг биелүүлнэ.
* Давталтын бие дотор тоолуурын утгыг өөрчилж болохгүй.
* Давталтын дотор бүхлээрээ багтсан өөр давталтыг бичиж болно.
* Зарчмын хувьд давталт төгсөхөөс өмнө гарч болно.

Тоолуурт давталтыг блок схемд дараах хэлбэрээр тэмдэглэнэ.

for(тоолуур=эхний утга, эцсийн утга, алхам) { тоолуурын бие }

**Жишээ 3.**  N натурал тооны факториалыг ол.

**{**  read(n);

p=1;

for(k=1,n,1) p=p\*k;

write(p);

**}**

**Жишээ 4.** Өгсөн n натурал тоо хүртэлх бүх анхны тоог ол.

Бодолт. ßмар нэг сондгой k - тоог анхны тоо мөн эсэхийг шалгахдаа 3- аас  хүртэлх сондгой тоонуудад хуваах хэрэгтэй. Хэрэв аль ч тоонд хуваагдахгүй бол анхны тоо мөн.

( Эсвэл сондгой k - тоог анхны тоо мөн эсэхийг шалгахдаа к хүртэлх бүх анхны тоонуудад хуваах хэрэгтэй.)

**{** ***// n – тоо 1 –ээс ихгүй тохиолдолд дахин оруулна.***

do{ read(n);

}while(n<=1);

write(“2”);

for(k=3,n,2)**{** ***// k- тоог анхны тоо эсэхийг шалгана.***

i=3;

t= sqrt(k);

while(k%i<>0 and i<=t) i=i+2;

if(i>t) write(k);

**}**

**}**

1. **Массивын тухай ойлголт**

Нэг ижил төрлийн олон тоог санах ойн дэс дараалсан үүрэнд хадгалж улмаар түүний элементүүдийг дугаараар нь олж боловсруулах хэрэгсэл программчлалын бүх хэлэнд байдаг. Ерөнхий нэртэй ижил төрлийн хувьсагчдыг хүснэгт хэмжигдэхүүн буюу массив гэнэ. Массивын элементүүдийг ялгах болон байрлалыг нь заахын тулд элементийн дугаарыг (индекс) хэрэглэнэ. Массивын элементүүд санах ойн дараалсан үүрүүдэд байрлана. Массивын нэр, элементийн төрөл, тоог тодорхойлж өгнө. Энэ үйлдлээр массивыг хадгалах санах ой хуваарилагдана.Массивын элементүүдэд утга олгох ба гаргахдаа элемент нэг бүрчлэн олгох ба гаргах үйлдлийг хийнэ. Массивыг дотор нь нэг хэмжээст болон олон хэмжээст массив гэж ангилна. Бид олон хэмжээст массиваар 2 хэмжээст дээр жишээ авч үзэх болно. Массивын индекс нь 0 –ээс эхэлнэ.

Ж:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 5 | 7 | 9 | 6 | 1 | 0 | 10 | 8 | 11 |
| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

***Олон хэмжээст массив:***

Олон хэмжээст массивыг зарлахдаа: эсвэл гэж зарлана. Олон хэмжээст массивд мөр баганын хувиар хандана. ()

**2 хэмжээст массивд анхны утга олгох**

2 хэмжээст массивд анхны утгыг олгохдоо 1хэмжээст массивд анхны утга олгодогтой төстэй буюу энэ массивыг 1хэмжээст массивуудын массив байдлаар ойлгогдохоор бичлэгтэй байдаг.

Олон хэмжээст массивд анхны утга олгохдоо (тэгш хэмт массивын хувьд):

;

}

}

Энэ нь мөрийн дагуу утга олгогдоно гэсэн үг. Жишээ: тэгш хэмт матрицын сондгой мөрийг 0-ээр, тэгш мөрийг 1-ээр дүүргэх алгоритм зохио.

{  
   
 }  
 {   
 }  
 }  
 }

}  
 }  
 }

1. **Дэд алгоритм (Функц)**

Тодорхой үр дүн агдаг эсвэл үйлдэл хийсэн биеэ даасан шинжтэй алгоритмыг тусд нь бичиж олон дахин ашиглах боломжтой алгоритмыг дэд алгоритм гэнэ. Дэд алгоритмуудыг ялгахын тулд нэрлэх бөгөөд үндсэн алгоритмаас дэд алгоритмыг дуудах, дэд алгоритмаас буцах гэсэн үйлдлүүдийг хэрэглэдэг. Үндсэн алгоритмаас дэд алгоритмд дамжуулах утгыг аргумент эсвэл параметр гэнэ. Харин дэд алгоритмаас үндсэн алгоритмд дамжуулах утгыг дэд алгоритмын үр дүн гэнэ. Дэд алгоритмыг нэрээр нь дуудаж хэрэглэнэ. Дэд алгоритм үндсэн алгоритмд утга буцааж болно, буцаахгүй ч байж болно. Үндсэн ба дэд алгоритмуудад бүгдэд нь хэрэглэгддэг хувьсагчийг **глобал** хувьсагч гэнэ. Харин тодорхой нэг дэд алгоритмд хэрэглэгдэх хувьсагчийг **локал** хувьсагч гэнэ.

Жишээ нь: натурал тоонуудын ХИЕХ-ийг олох

Дэд алгоритм нь:

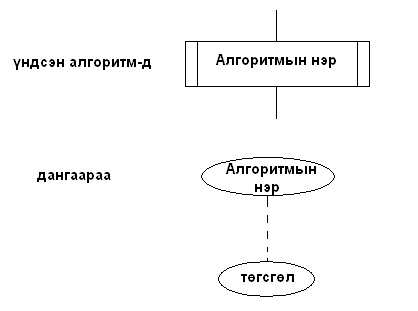
}

}

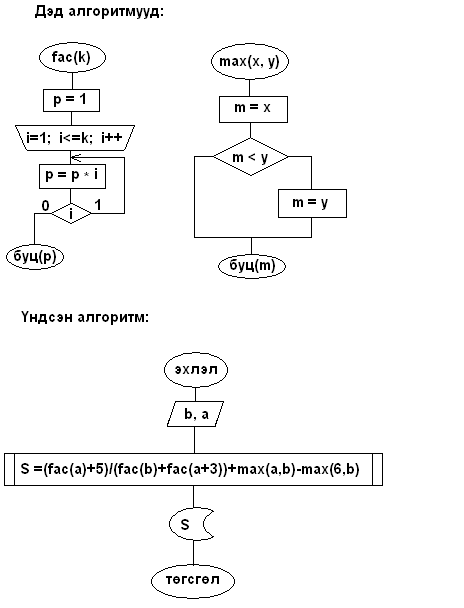
Үндсэн алгоритм нь:

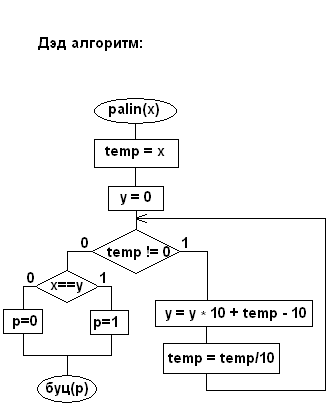
}

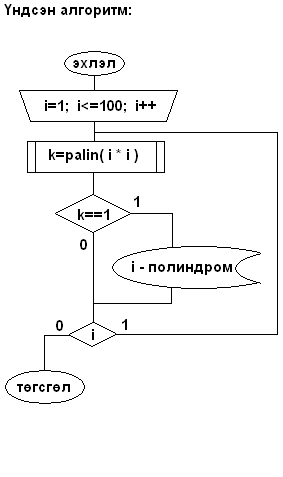
Дэд алгоритмыг блок схемд дараах байдлаар тэмдэглэнэ.

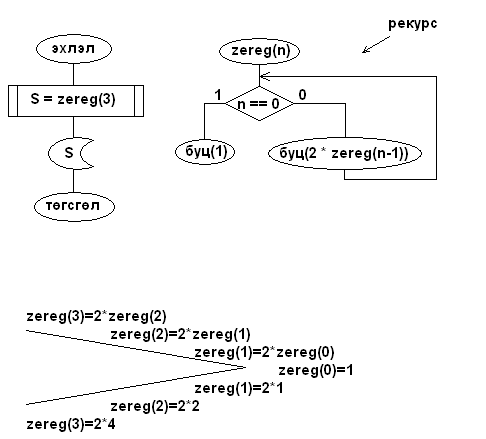


Жишээ:



Бодлого:Квадрат зэрэг нь полиндром байх бүх тоог ол.

****

****

**Рекурс**

Өөрөө өөрийгөө дуудсан функцыг рекурсив функц гэнэ. Рекурсив нь зогсох нөхцөл хүртэл өөрийгөө дуудан ажиллаад дараа нь дуудагдсан дарааллаараа гэдрэг буцна.

Жишээ: -ийг ол.

Жишээ: тооны факториал олох алгоритмыг рекурс ашиглан бич.

