Publicerad 2007-02-15 18:21 av PolyMathWolverine

reccurence rekations is the anstract structure of life

permutations in evolution

```
#include
#include
#define MAX_P 25

struct permutation
{
int length;
int number;
int perm[MAX_P];
struct permutation *next;
};

struct permutation *permutation_list;

struct permutation* build_perm_list(struct permutation*,int n);
void print_perm_list(struct permutation*);

int main(int argc, char *argv[])
{
int n;
struct permutation *ny;
ny=(struct permutation*)malloc(sizeof(struct permutation));
ny->length=1;
ny->next=NULL;
ny->number=1;
ny->perm[0]=1;
permutation_list=ny;

printf("size of permutation <=n :");
scanf("%d",&n);
permutation_list=build_perm_list(permutation_list,n);
print_perm_list(permutation_list);

system("PAUSE");
return 0;
}

struct permutation* build_perm_list(struct permutation *first,int n)
```

```
{
struct permutation *ny;
int i,j,l;

if(n==1)
return first;
else if(n>=2)
{
for(l=2;l<=n;l++)
{
/*i representerar platsen där vi ska placera ut l*/
for(i=0;i<=l-1;i++)
{ /*skapa en ny permutation av längd l
från den som first pekar på av längd l-1*/
ny =(struct permutation*)malloc(sizeof(struct permutation));
ny->length=first->length+1;
ny->number=first->number+1;
ny->next=NULL;
j=0;
while(j {
ny->perm[j]=first->perm[j];/*kopiering*/
j=j+1;
}
ny->perm[j]=l;
j=j+1;
while(j<=l-1);
{
ny->perm[j]=first->perm[j];/*kopiering*/
j=j+1;
}
/*inlänkningsdags*/
ny->next=first;
first=ny;
}
}
return first;
}
else
{
return NULL;
}
}

void print_perm_list(struct permutation *first)
{
int i;
struct permutation *tmp;
tmp=first;
while(tmp!=NULL)
{
printf("np[%d] ",tmp->number);
for(i=0;i<=tmp->length-1;i++)
{
printf("%d",tmp->perm[i]);
}
printf("n");
tmp=tmp->next;
}

}
```