

contrôle d'informatique

Qu'écrit le programme suivant quand on l'exécute ?

```
#include<stdio.h>
void pr(int a,int b)
{ if(b>0) pr(a+3,b-1),pr(a+2,b-2),pr(a+4,b-1);
  else printf("%4d",a);
}
void pr2(int a,int b)
{ printf("%4d",a);
  while(b) pr2(a++,--b);
}
#define a
#define b
int main()
{ int i,j,k,l,*p=&i,*q=&j,*r;
  printf("%4d %4d %4d %4d\n",a&b,a|b,a^b,(a,b));
  printf("%4d %4d %4d %4d\n",-a,~a,!a,!!a);
  printf("%4d %4d %4d %4d\n",i=a,j=b,k=a+b,l=a-b);
  printf("%4d %4d %4d %4d\n",++i*j--,k+=l--,a<<2,a>>2);
  printf("%4d %4d %4d %4d\n",++i*j--,k+=l--,a<<2,a>>2);
  r=p,p=q,q=r;
  i=a,j=b;
  printf("%4d %4d %4d %4d\n",i,j,*p,*q);
  j+=i+*q, j+=i+*q;
  printf("%4d %4d %4d %4d\n",i,j,*p,*q);
  pr (a,3); printf("\n");
  pr2(a,3); printf("\n");
  return 0;
}
```

Compléter les deux procédures

```
liste select(liste a,int masque);
```

```
liste tri(liste a):
```

qui permutent les chaînons d'une liste chaînée. `select(a,masque)` met en tête tous les chaînons dont la clé `v` vérifie `v&masque==0`. Tous les autres chaînons suivent. Les chaînons gardent l'ordre de la liste initiale s'il le peuvent. `tri()` trie son argument dans l'ordre croissant des clés en utilisant `select`.

barème sur 22.58 points

Les 7 printf du programme principal affichent chacun 4 nombres. Cela fait en tout 28 nombres qui comptent chacun 0.33 point. $28/3=9.33$ points

pr(a,3) affiche 17 nombres qui comptent chacun 0.25 point. $17/4=4.25$ points

pr2(a,3) affiche 8 nombres qui comptent chacun 0.25 point. $8/4=2$ points

select() 4.5 points

tri() 2.5 points

```
#include<stdio.h>
#include<stdlib.h>
typedef struct chainon *liste;
struct chainon{int val; liste suite;};
void aff(liste a)
{ for(;a;a->suite) printf("%d ",a->val);
  printf("\n");
}
liste selectm(liste a,int masque)
{ liste p=0, q=0, r; // 0.5pt
  while(a) // while 0.5pt, if 0.5pt
    if(a->val&masque) r=a, a=r->suite, r->suite=p, p=r; // 0.5pt
    else r=a, a=r->suite, r->suite=q, q=r; // 0.5pt
  while(p) r=p, p=r->suite, r->suite=a, a=r; // while 0.5pt, 0.5pt
  while(q) r=q, q=r->suite, r->suite=a, a=r; // while 0.5pt, 0.5pt
  return a;
}
liste tri(liste a)
{ int masque;
  for(masque=1;masque;masque<<=1) // 1 point
    a=selectm(a,masque); // 1 point
  return a; // 0.5 point
}
liste nouv(int val,liste suite)
{ liste a=malloc(sizeof(*a));
  a->val=val;
  a->suite=suite;
  return a;
}
int main()
{ liste a=0; int i;
  for(i=0;i<10;i++) a=nouv(i*3%10,a);
  aff(a);
  a=tri(a);
  aff(a);
  return 0;
}
```