

Exercice 1

1. typedef struct{
 int Operande1, Operande2 ;
 char Operateur ;
 int Resultat } Operation ;
2. void Saisir_Operation(Operation *op)
 { scanf(« %d%d%c », &op->Operande1, &op->Operande2, &op-> Operateur) ;
 }
3. void Afficher_Operation(Operation op)
 { printf(« %d%c%d=%d »,op.Operande1, op.Operateur, op.Operande2,o p.Resultat) ;
 }
4. void Calculer_Operation(Operation *op)
 { int r ;
 Switch(op->Operateur){
 case ('+') : r= op->Operande1+ op->Operande2 ; break ;
 case ('-') : r= op->Operande1- op->Operande2 ; break ;
 case ('*') : r= op->Operande1* op->Operande2 ; break ;
 case ('/') : r= op->Operande1/ op->Operande2 ; break ;
 case ('%') : r= op->Operande1% op->Operande2 ; break ;
 }
 Op->Resultat= r;
 }
5. void Calculer(int N)
 {
 Operation * top=(Operation *)malloc(N*sizeof(Operation));
 for (Operation* p =top; p<top+N; p++)
 {
 Saisir_Operation(p);
 Calculer_Operation(p) ;
 Afficher_Operation(*p) ;
 }
 }

Exercice 2

```
1. typedef struct{
    char   Code_banque      [5];
    char   Code_guichet     [5];
    char   Numero_client    [11];
    char   Clef_RIB         [2];
} Numero_Compte;

2. void Saisir_Compte(Numero_Compte *nc)
{scanf("%s %s %s %s",&nc->Code_banque,&nc->Code_guichet,&nc->Numero_client,&nc->Clef_RIB);
}

3. char Convertir_Caractere(char c)
{
char h;
switch (c)
{
    case '0': h='0';break;
    case '1':;
    case 'A':;
    case 'J': h='1';break;
    case '2':;
    case 'B':;
    case 'K':;
        case 'S': h='2'; break;
    }
return h;
}

4. void Convertir_Compte (Numero_Compte nc, char cc[])
{
int i,j=0;
for (i=0;i<5;i++)
{
    cc[j]=Convertir_Caractere(nc.Code_banque[i]);
    j++;
}
for (i=0;i<5;i++)
{
    cc[j]=Convertir_Caractere(nc.Code_guichet[i]);
    j++;
}
for (i=0;i<11;i++)
{
    cc[j]=Convertir_Caractere(nc.Numero_client[i]);
```

```

    j++;
}
for (i=0;i<2;i++)
{
    cc[j]=Convertir_Caractere(nc.Clef_RIB[i]);
    j++;
}
cc[23]='\0';
puts(cc);
}

```

```

5. void Valider_Compte(Numero_Compte *nc)
{
    char s[24];
    long l;
    Saisir_Compte(nc);
    Convertir_Compte(*nc,s);
    puts(s);
    l=atol(s);
    if (l%97)
        puts("Numéro de compte non valide ");
    else
        puts("Numéro de compte valide ");
}

```

```

6. int main(int argc, char *argv[])
{
    Numero_Compte N;
    Valider_Compte(&N);
}

```