```
resimilarly the student SA is togethered for the classes Circins
Polynomial differentiation
Rolling Ordanolum:
Void diff()
                                              if (play - exp > play -> exp)
 Poly * pln 1. * new node;
                                               rewrode > co引:ptx, -> co引;
 Phi - List 1:
while (ptr ! = Null)
                                               newnode -> eup = pti, -> eup;
                                              newrode -> next = NOLL,
neumale - malloc ( &ize of (sturt poly));
                                               list 3 = Create ( list 3, new mode);
nownade - coeff = ptr 1 - coeff * ptr -> exp;
                                             Plai - plai - nent;
new node -> exp = ptr1 -> exp-1;
newrode -> nent = NULL;
                                             elle
List = create (list 8, new node);
pte 1 = pte > nent i
                                              new → weff = ptr 2 → weff;
                                              new mad -> emp = pth = -> emp;
Routine:
                                              newnode > nent = NULL;
Void Sub()
                                              list 3 = create (list 3, new node);
 Poly * ptr, * ptr2, *neworade
                                             ptiz = ptiz -> next;
 Physiki:
 Pthe = Linte;
 while (ptr,!=NULL ||Ptre!=NULL)
  nevarade = Malloc (Liee of (8thurt poly));
  if (ptr1 > exp = = ptr2 -> exp)
                                                    Addition
 rewrode -> coeff = (ptr -> coeff) - (ptr2 > coeff); // newrode -> coeff = (ptr -> coeff)
  new node -> enp = ptr 1 -> enp:
  hewhode > nent = NULL;
  list 3 = Create (Isst 3, newwode);
  Ptr = ptr > next;
Ptre = ptre > next;
```