Help

```
#include "integral.h"
#include <stddef.h>
#include <stdlib.h>

#define FUNC(x) ((*func)(x))
#define FUNK(x) (2.0*(x)*(*func2)(aa+(x)*(x)))
#define FUNKY(x) ((*func1)(1.0/(x))/((x)*(x)))

#define NR_END 1
#define FREE_ARG char*
double *vector(long nl,long nh){
  double *v;

  v=(double *)malloc((size_t)((nh-nl+1+NR_END)*si
    zeof(double)));
  /*if(!v) callerror("allocation failure in vect
    or()");*/
  return v-nl+NR_END;
}

void free_vector(double *v,long nl,long nh){
  free((FREE_ARG)(v+nl-NR_END));
}


double midpnt(double (*func)(double), double a,
    double b, int n){
  double x,tnm,sum,del,ddel;
  static double s;
  int it,j;

  if(n==1){
    s=(b-a)*FUNC(0.5*(a+b));
  return s;
  } else {
  for(it=1,j=1;j<n-1;j++) it*=3;
  tnm=it;
  del=(b-a)/(3.0*tnm);
  ddel=del+del;
  x=a+0.5*del;
```

```
  sum=0.0;
  for(j=1;j<=it;j++) {
    sum+=FUNC(x);
    x+=ddel;
    sum+=FUNC(x);
    x+=del;

  }
  s=((b-a)*sum/tnm)/2.0;

  return s;
  }
}

double midpntbis(double (*func)(double), double
    a, double b, int n){
  double x,tnm,sum,del,ddel;
  static double s;
  int it,j;

  if(n==1){
      s=(b-a)*FUNC(0.5*(a+b));
  return s;

  } else {
  for(it=1,j=1;j<n-1;j++) it*=3;
  tnm=it;
  del=(b-a)/(3.0*tnm);
  ddel=del+del;
  x=a+0.5*del;
  sum=0.0;
  for(j=1;j<=it;j++){
    sum+=FUNC(x);
    x+=ddel;
    sum+=FUNC(x);
    x+=del;
  }
  s=((b-a)*sum/tnm)/2.0;
  return s;
  }
}
```

```
double midsql(double (*func2)(double), double aa,
     double bb, int n){
  double x,tnm,sum,del,ddel,b,a;
  static double s;
  int it,j;

  b=sqrt(bb-aa);
  a=0.0;
  if(n==1) {
         s=(b-a)*FUNK(0.5*(a+b));
  return s;
  } else {
  for(it=1,j=1;j<n-1;j++) it*=3;
  tnm=it;
  del=(b-a)/(3.0*tnm);
  ddel=del+del;
  x=a+0.5*del;
  sum=0.0;
  for(j=1;j<=it;j++){
    sum+=FUNK(x);
    x+=ddel;
    sum+=FUNK(x);
    x+=del;
  }
  s=((b-a)*sum/tnm)/2.0;
  return s;
  }
}


double midsqlbis(double (*func2)(double), double
    aa, double bb, int n){
  double x,tnm,sum,del,ddel,b,a;
  static double s;
  int it,j;

  b=sqrt(bb-aa);
  a=0.0;
  if(n==1) {
    s=(b-a)*FUNK(0.5*(a+b));
```

```
  return s;
  } else {
  for(it=1,j=1;j<n-1;j++) it*=3;
  tnm=it;
  del=(b-a)/(3.0*tnm);
  ddel=del+del;
  x=a+0.5*del;
  sum=0.0;
  for(j=1;j<=it;j++) {
    sum+=FUNK(x);
    x+=ddel;
    sum+=FUNK(x);
    x+=del;
  }
  s=((b-a)*sum/tnm)/2.0;
  return s;
  }
}

double midinf(double (*func1)(double), double aa,
     double bb, int n){
  double x,tnm,sum,del,ddel,b,a;
  static double s;
  int it,j;

  b=1.0/aa;
  a=1.0/bb;
  if(n==1){
    s=(b-a)*FUNKY(0.5*(a+b));
    return s;
  } else {
  for(it=1,j=1;j<=n-1;j++) it*=3;
  tnm=it;
  del=(b-a)/(3.0*tnm);
  ddel=del+del;
  x=a+0.5*del;
  sum=0.0;
  for(j=1;j<=it;j++){
    sum+=FUNKY(x);
    x+=ddel;
    sum+=FUNKY(x);
```

```
    x+=del;
  }
  s=((b-a)*sum/tnm)/2.0;
  return s;
  }
}
```

References