Problem 1

C code:

```
#include<stdio.h>

void dgesv_(int *n, int *NRHS, double *A, int *LDA,
            int *IPIV, double *B, int *LDB, int *INFO);

void dgemm_(char *TRANSA, char *TRANSB, int *M, int *n, int *K,
            double *ALPHA, double *A, int *LDA, double *B,
            int *LDB, double *BETA, double *C, int *LDC);

int main(int argc, char *argv[]){
        FILE *fp;
        int N=0, P1=1,P,i,j,k;
        char tmp;
        fp = fopen(argv[1],"r");
        while((tmp=fgetc(fp))!=EOF){
                if(tmp=='\n')    N++;
        }
        rewind(fp);
        while((tmp=fgetc(fp))!= '\n'){
              if(tmp==' ') P1++;
        }
        rewind(fp);
        printf("Sample size and number of predictors are %d and %d respective

        double Y[N];
        if(atoi(argv[2])==1) P = P1;
        else P = P1 - 1;
        double X[N][P];
        for(i=0;i<N;i++){
                for(j=0;j<P1;j++){
                        if(j==0){
                                fscanf(fp, "%lf", &Y[i]);
                                if(atoi(argv[2])==1) X[i][0] = 1;
                        }
                        else{
                                if(atoi(argv[2])==1)
                                        fscanf(fp, "%lf", &X[i][j]);
                                else fscanf(fp, "%lf", &X[i][j-1]);
```

Problem 1 continued on next page. . .

```
                    }
            }
    }

    double x[N*P], xt[P*N], Design_Mat[P*P];
    int n1=N, n2=P, n3=1, info, ipiv[P];
    double coef[P], beta=0, alpha=1;
    char TRANSA = 'N', TRANSB='N';

    for(i=0;i<N;i++){
            for(j=0;j<P;j++){
                    x[i*P+j] = X[i][j];
                    xt[j*N+i] = X[i][j];
            }
    }

    dgemm_(&TRANSA, &TRANSB, &n2, &n2, &n1, &alpha,
                    x, &n2, xt, &n1, &beta, Design_Mat, &n2);
    dgemm_(&TRANSA, &TRANSB, &n2, &n3, &n1, &alpha,
                    x, &n2, Y, &n1, &beta, coef, &n2);
    dgesv_(&n2,&n3, Design_Mat,&n2, ipiv, coef,&n2,&info);

    printf("The regression coefficients: ");
    for(i=0;i<P;i++)
            printf("%f  ", coef[i]);
    printf("\n");
    return(0);
}
```

Problem 2

(a)

$$\int_0^\infty (x^2+5)xe^{-x}dx \approx 10.99763 \pm 0.2028816$$

(b)

$$\int_0^1 \int_{-\infty}^\infty e^{-x^2}\cos(xy)dxdy \approx 1.634335$$

Problem 2 continued on next page. . .

(c)

$$\int_0^\infty \frac{3}{4}x^4 e^{-x^3/4}dx \approx 2.271137 \pm 0.00957$$

R code:

```
## (a)
N <- 100000
set.seed(20170311)
rand_exp<- rexp(N, rate = 1)
mean((rand_exp^2+5)*rand_exp)
sqrt(var((rand_exp^2+5)*rand_exp)/N)*1.96

## (b)
set.seed(20170311)
rand_norm <- rnorm(N, mean = 0, sd = 1/sqrt(2))
rand_unif <- runif(N, min = 0, max = 1)
mean(cos(rand_norm*rand_unif)*sqrt(pi))

## (c)
set.seed(20170311)
rand_weibull <- rweibull(N, shape = 3, scale = 4^(1/3))
mean(rand_weibull^2)
sqrt(var(rand_weibull^2)/N)*1.96
```

Problem 3

- $\nu = 0.1$: $I \approx 0.185114 \pm 0.07967168$;
- $\nu = 1$: $I \approx 0.1363114 \pm 0.00121565$;
- $\nu = 10$: $I \approx 0.1386195 \pm 0.004512526$.

R code:

```
##
v = 0.1
N <- 100000
set.seed(20170311)
rand_norm <- rnorm(N, mean = 1.5, sd = v)
mean(v*exp(-rand_norm^2/2)/exp(-(rand_norm-1.5)^2/(2*v^(2)))*
      ifelse(rand_norm<2&rand_norm>1,1,0))
```

```
sqrt(var(v*exp(-rand_norm^2/2)/exp(-(rand_norm-1.5)^2/(2*v^(2)))*
        ifelse(rand_norm<2&rand_norm>1,1,0))/N)*1.96

## v = 1
v = 1
set.seed(20170311)
rand_norm <- rnorm(N, mean = 1.5, sd = v)
mean(v*exp(-rand_norm^2/2)/exp(-(rand_norm-1.5)^2/(2*v^(2)))*
        ifelse(rand_norm<2&rand_norm>1,1,0))
sqrt(var(v*exp(-rand_norm^2/2)/exp(-(rand_norm-1.5)^2/(2*v^(2)))*
            ifelse(rand_norm<2&rand_norm>1,1,0))/N)*1.96

## v = 10
v = 10
set.seed(20170311)
rand_norm <- rnorm(N, mean = 1.5, sd = v)
mean(v*exp(-rand_norm^2/2)/exp(-(rand_norm-1.5)^2/(2*v^(2)))*
        ifelse(rand_norm<2&rand_norm>1,1,0))
sqrt(var(v*exp(-rand_norm^2/2)/exp(-(rand_norm-1.5)^2/(2*v^(2)))*
            ifelse(rand_norm<2&rand_norm>1,1,0))/N)*1.96
```

Problem 4

(a) $\hat{I} = 0.6977335$
(b) $E\{c(U)\} = 1.5$. The optimal value of $b = -0.4779221$. $\hat{I}_{CV} = 0.6927941$.
(c)

$$\widehat{var}(\hat{I}_{MC}) = 1.295893 \times 10^{-5} > \widehat{var}(\hat{I}_{CV}) = 4.006056 \times 10^{-7}$$

(d) We define a new function $c_1(x) = \sqrt{1+x}$. The new estimator is denoted by

$$\hat{I}_{new} = \frac{1}{n}\sum_{i=1}^{n} h(U_i) - b\left[\frac{1}{n}\sum_{i=1}^{n} c_1(U_i) - E\{c_1(U)\}\right].$$

The variance of $\hat{I}_{new}$ is smaller than the variance of $\hat{I}_{CV}$ because

$$\begin{aligned}
\left|\text{Corr}\left(\frac{1}{1+x}, \sqrt{1+x}\right)\right| &= \frac{E\frac{1}{\sqrt{1+x}} - E\frac{1}{1+x}E\sqrt{1+x}}{\sqrt{var(\frac{1}{1+x})var(\sqrt{1+x})}} = 0.990998 \\
\left|\text{Corr}\left(\frac{1}{1+x}, 1+x\right)\right| &= \frac{E1 - E\frac{1}{1+x}E(1+x)}{\sqrt{var(\frac{1}{1+x})var(1+x)}} = 0.9841661 < 0.990998
\end{aligned}$$

R code:

```r
## (a)
n = 1500
set.seed(20170311)
rand_unif <- runif(n, 0, 1)
mean(1/(1+rand_unif))

## (b)
b_opt <- sum((1/(1+rand_unif)- mean(1/(1+rand_unif)))*
                  ((1+rand_unif)-mean((1+rand_unif))))/
                  (var(1+rand_unif)*(n-1))

mean(1/(1+rand_unif)) - b_opt*(mean(1+rand_unif) - 1.5)

## (c)
var(1/(1+rand_unif))/n
var(1/(1+rand_unif)-b_opt*(1+rand_unif))/n

## (d)
(1-log(2)*1.5)/((1/2-log(2)^2)/12)^(1/2)
(2*sqrt(2)-2-log(2)*((4*sqrt(2)-2)/3))/
  ((1/2-log(2)^2)*(1.5-((4*sqrt(2)-2)/3)^2))^(1/2)
```