Hamming Code in all 4 possible consequences:

Even Parity (Data word to Code Word):

Code:

#include <stdio.h>

int main()

{

int a,b,c,d;

int i;

int data[7];

printf("Data Word:");

scanf("%d %d %d %d",&a,&b,&c,&d);

data[0] = a;

data[1] = b;

data[2] = c;

data[4] = d;

int size = sizeof(data)/sizeof(data[0]);

printf("Data Word : %d %d %d %d\n",data[0],data[1],data[2],data[4]);

data[6] = data[4]^data[2]^data[0];

data[5] = data[4]^data[1]^data[0];

data[3] = data[2]^data[1]^data[0];

printf("Hamming Code Word : ");

for(i=0;i<size;i++)

{

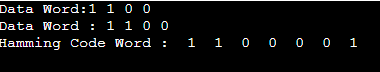
printf(" %d ",data[i]);

}

return 0;

}

Output:



Odd Parity(Data word to Code Word):

Code:

#include <stdio.h>

int main()

{

int a,b,c,d;

int i;

int data[7];

printf("Data Word:");

scanf("%d %d %d %d",&a,&b,&c,&d);

data[0] = a;

data[1] = b;

data[2] = c;

data[4] = d;

int size = sizeof(data)/sizeof(data[0]);

printf("Data Word : %d %d %d %d\n",data[0],data[1],data[2],data[4]);

data[6] = !(data[4]^data[2]^data[0]);

data[5] = !(data[4]^data[1]^data[0]);

data[3] = !(data[2]^data[1]^data[0]);

printf("Hamming Code Word : ");

for(i=0;i<size;i++)

{

printf(" %d ",data[i]);

}

return 0;

}

Output:



Even Parity(Error Code to corrected code):

Code:

#include <stdio.h>

int main()

{

int data[7];

int i;

int size = sizeof(data)/sizeof(data[0]);

printf("Enter the error code : ");

for(i=0;i<size;i++)

{

scanf("%d",&data[i]);

}

/\*

d[0] = d7

d[1] = d6

d[2] = d5

d[3] = p4

d[4] = d3

d[5] = p2

d[6] = p1

\*/

int c1 = data[6]^data[4]^data[2]^data[0];

int c2 = data[5]^data[4]^data[1]^data[0];

int c4 = data[3]^data[2]^data[1]^data[0];

// location will be calculated as c4 c2 c1 order

int error = c4\*4+c2\*2+c1\*1;

printf("\nError at location : %d\n",error);

data[size-error] = !(data[size-error]);

printf("After fixing the error \n");

for(i=0;i<size;i++)

{

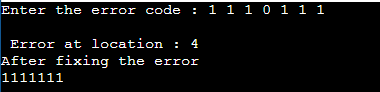
printf("%d",data[i]);

}

return 0;

}

Output:



Odd Parity(Error Code to corrected code):

Code:

#include <stdio.h>

int main()

{

int data[7];

int i;

int size = sizeof(data)/sizeof(data[0]);

printf("Enter the error code : ");

for(i=0;i<size;i++)

{

scanf("%d",&data[i]);

}

/\*

d[0] = d7

d[1] = d6

d[2] = d5

d[3] = p4

d[4] = d3

d[5] = p2

d[6] = p1

\*/

int c1 = !(data[6]^data[4]^data[2]^data[0]);

int c2 = !(data[5]^data[4]^data[1]^data[0]);

int c4 = !(data[3]^data[2]^data[1]^data[0]);

// location will be calculated as c4 c2 c1 order

int error = c4\*4+c2\*2+c1\*1;

printf("\nError at location : %d\n",error);

data[size-error] = !(data[size-error]);

printf("After fixing the error \n");

for(i=0;i<size;i++)

{

printf("%d",data[i]);

}

return 0;

}

Output:

