TABLE II: Our Defect Class Classification and Examples of Each Defect Class

AST Type	Defect Type	Defect Class	Example
Statement	Control flow	(SDIF) Delete if, else, else if, for or while	– if (lines[i].y1 == last->y1)
		(SIIF) Insert if, else, else if, for or while	+ if(*l*)
		(SRIF) Replace if, else, else if, for or while	– if(a==b) + if(mask(a)==b)
		(SDRT) Delete return	– return 0;
		(SIRT) Insert return	+ return 0;
		(SDIB) Delete/Insert break or continue	– break;
	Data flow	(SDLA) Delete assignment	– answer+=((i-1)*dif);
		(SISA) Insert assignment	+ t=0;
	Function call	(SDFN) Delete function call	– printf(“%s %s\n”,s1,s2);
		(SISF) Insert function call	+ scanf(“%d”, &n);
	Type	(STYP) Replace variable declaration type	– int a; + long a;
	Move	(SMOV) Move statement	– scanf(“%d”,&i); scanf(“%s”, &a); + scanf(“%d”,&i);
		(SMVB) Move brace up/down	– } printf(“%d”,c); + }
Operator	Control flow	(ORRN) Replace relational operator	– if(sum>n) + if(sum>=n)
		(OLLN) Replace logical operator	– if((s[i] == ’4’) && (s[i] == ’7’)) + if((s[i] == ’4’) || (s[i] == ’7’))
		(OILN) Insert && (tighten condition) or || (loosen condition)	– if(t%2==0) + if(t%2==0 && t!=2)
		(OEDE) Replace = with == or vice versa	– else if(n=1 && k==1) + else if(n==1 && k==1)
		(OICD) Insert a conditional operator	– printf (“%d\n”, i); + printf (“%d\n”, 3 == x ? 5 : i);
	Arithmetic	(OAAN) Replace arithmetic operator	– v2-=d; + v2+=d;
		(OAIS) Insert/Delete arithmetic operator	– max += days%2; + max += (days%7)%2;
		(OAID) Insert/Delete/Replace $++$ or $--$	+ i++;
		(OMOP) Modify operator precedence	– ans=max(ans,l-arr[n]*2); + ans=max(ans,(l-arr[n])*2);
	Function call	(OFFN) Alternative function call	– fflush(stdin); + getchar();
		(OFPF) Replace print format	– printf(“%d\n”,l); + printf(“%lld\n”,l);
		(OFPO) Modify function parameter order	– if(strcmp(c[i],b)>0) + if(strcmp(b,c[i])>0)
	Pointer	(OIRO) Insert/Delete Reference Operator	– printf(“%d”,&t); + printf(“%d”,t);
	Type	(OITC) Insert type cast operator	– ((p2m/p1m)*t+1 + ((float)p2m/p1m)*t+1;
Operan**D**	Constant	(DCCR) Replace constant with variable/constant	– for(i=n+1;i<=9000;i++) + for(i=n+1;i<=10000;i++)
	Variable	(DRVA) Replace a read variable with a variable/constant	– for (i=0;i<l;i++) + for (i=0;i<m;i++)
		(DRWV) Replace a write variable with a variable	– b=0; + a=0;
	Array	(DMAA) Insert/Replace array access	– out[l] = ’\0’; + out[l−−] = ’\0’;
		(DRAC) Replace constant of array initialization	– int ex[2]={0,2}; + int ex[2]={0,3};
		(DCCA) Modify array size	– int x[100] + int x[100000];
Higher order	Non-branch stmt	(HDMS) Delete multiple non-branch statements	– freopen(“input.txt”, “r”, stdin); – freopen(“output.txt”, “w”, stdout);
		(HIMS) Insert multiple non-branch statements	+ freopen(“input.txt”, “r”, stdin); + freopen(“output.txt”, “w”, stdout);
		(HDIM) Delete and insert multiple non-branch statements	– break; + count=0;
	Branch stmt	(HBRN) Delete/Insert branch statement and non-branch statements	+ if(len%slov!=0){printf(“NO”); + return 0;}
	Expressions	(HEXP) Delete/Insert/Replace multiple operators and operands	–if(m*9>=s && s) +if((m*9>=s && s) || (m==1 && !s))
	Combination	(HCOM) Insert/Replace statements and expressions	– rep(i,n) + for(i=n-1;i>=0;i−−)
	Others	(HOTH) Other higher order defect classes	– scanf(“%s”,h); + for(i=0;i<71;i++) + scanf(“%c”,&h[i]);